

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-114469

(43) 公開日 平成7年(1995)5月2日

(51) Int.Cl. ⁶	識別記号	片内整理番号	F I	技術表示箇所
G 0 6 F 9/34	3 2 0 A			
9/30	3 5 0 B			

審査請求 未請求 請求項の数 4 O L (全 76 頁)

(21) 出願番号 特願平5-260199

(22) 出願日 平成5年(1993)10月18日

(71) 出願人 000006013

三菱電機株式会社

東京都千代田区丸の内二丁目2番3号

(72) 発明者 佐藤 貢

兵庫県伊丹市瑞原4丁目1番地 三菱電機

株式会社エル・エス・アイ研究所内

(72) 発明者 吉田 豊彦

兵庫県伊丹市瑞原4丁目1番地 三菱電機

株式会社エル・エス・アイ研究所内

(72) 発明者 岩田 俊一

兵庫県伊丹市瑞原4丁目1番地 三菱電機

株式会社エル・エス・アイ研究所内

(74) 代理人 弁理士 高田 守

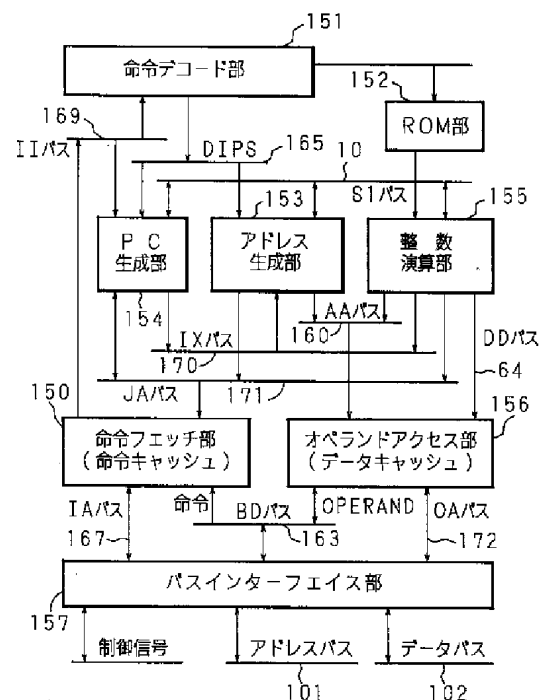
(54) 【発明の名称】 データ処理装置

(57) 【要約】

【目的】 バーストモードによるブロック転送機能により高速なバスアクセスを行ない、ストリング操作命令及びビットマップ操作命令を効率よく実行し、データ処理装置の性能を向上させる。

【構成】 ストリング操作命令及びビットマップ操作命令を有し、バースト転送機能でデータを入出力するバスインターフェイス部157と、主/副の二つのALUを内蔵した整数演算部155とを備え、データバス102の幅より大きい単位でデータバス102を通じて外部メモリとの間でデータを送受する処理を反復して実行する。

【効果】 安価で低速なメモリシステムが接続されている場合にも、ストリングデータ、ビットマップデータを高速処理するデータ処理装置が得られる。



【特許請求の範囲】

【請求項 1】 第 1 のビット幅のデータバスに接続された外部メモリと、
 命令をデコードする命令デコーダと、
 前記命令デコーダが命令をデコードした結果を示す信号が与えられることにより、命令の実行を制御するための制御信号を発生する実行制御部と、
 実行すべき命令の処理対象のデータを保持するデータレジスタと、前記処理対象のデータの前記外部メモリにおけるアドレスを保持するアドレスレジスタと、前記第 1 のビット幅より大きいまたは等しい第 2 のビット幅を処理単位として演算を施す少なくとも一つの演算器とを有し、前記実行制御部が出力する制御信号に従って命令を実行する命令実行部と、
 前記バスを通じて 1 回のアドレス出力に対して n (n は 2 以上の整数) 回のデータ転送を行なうことにより、前記第 2 のビット幅より大きい第 3 のビット幅のデータをバースト転送して前記命令の実行に必要なデータを前記外部メモリとの間で入出力するバスインタフェイス部とを備えたデータ処理装置において、
 前記命令デコーダが、前記外部メモリの第 1 のメモリ領域に連続して格納されている第 1 の多ビットデータ列と前記外部メモリの第 2 のメモリ領域に連続して格納されている第 2 の多ビットデータ列との各ビット間で所定の演算を順次施した結果の各ビットを前記第 1 または第 2 のメモリ領域に格納する多ビットデータ列演算命令をデコードした場合に、
 前記実行制御部は、
 前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記第 1 及び第 2 のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記第 1 及び第 2 のメモリ領域に格納されている第 1 及び第 2 の多ビットデータ列をそれぞれ第 3 のビット幅分だけ前記外部メモリからバースト転送で読み出して前記データレジスタに格納させる第 1 の制御と、
 前記命令デコーダから与えられる信号に基づいて前記所定の演算を指示する信号を与えることにより、前記演算器に、前記第 2 のビット幅を処理単位として、前記データレジスタに格納された第 3 のビット幅の第 1 のデータ列と第 2 のデータ列との間で前記所定の演算を施して演算結果を第 3 のビット幅の第 3 のデータ列として前記データレジスタに再度格納させる第 2 の制御と、
 前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記第 1 または第 2 のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記データレジスタに格納された前記第 3 のビッ

ト幅の第 3 のデータ列をバースト転送で前記第 1 または第 2 のメモリ領域に格納させる第 3 の制御と、
 前記命令デコーダから与えられる信号に基づいて前記第 1、第 2 及び第 3 の制御を所定回数反復することにより、前記第 3 のデータ列を前記第 1 または第 2 のメモリ領域に連続的に格納する第 4 の制御とを実行すべくしてあることを特徴とするデータ処理装置。

【請求項 2】 第 1 のビット幅のデータバスに接続された外部メモリと、
 命令をデコードする命令デコーダと、
 前記命令デコーダが命令をデコードした結果を示す信号が与えられることにより、命令の実行を制御するための制御信号を発生する実行制御部と、
 実行すべき命令の処理対象のデータを保持するデータレジスタと、前記処理対象のデータの前記外部メモリにおけるアドレスを保持するアドレスレジスタと、前記第 1 のビット幅より大きいまたは等しい第 2 のビット幅を処理単位として演算を施す少なくとも一つの演算器とを有し、前記実行制御部が出力する制御信号に従って命令を実行する命令実行部と、
 前記バスを通じて 1 回のアドレス出力に対して n (n は 2 以上の整数) 回のデータ転送を行なうことにより、前記第 2 のビット幅より大きい第 3 のビット幅のデータをバースト転送して前記命令の実行に必要なデータを前記外部メモリとの間で入出力するバスインタフェイス部とを備えたデータ処理装置において、
 前記命令デコーダが、前記外部メモリの第 1 のメモリ領域に連続して格納されている多ビットデータ列に所定の演算を順次施した結果の各ビットを前記第 1 のメモリ領域または前記第 1 のメモリ領域とは異なる第 2 のメモリ領域に格納する多ビットデータ列演算命令をデコードした場合に、
 前記実行制御部は、
 前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記第 1 のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記第 1 のメモリ領域に格納されている多ビットデータ列を前記第 3 のビット幅分だけ前記外部メモリからバースト転送で読み出して前記データレジスタに格納させる第 1 の制御と、
 前記命令デコーダから与えられる信号に基づいて前記所定の演算を指示する信号を与えることにより、前記演算器に、前記第 3 のビット幅を処理単位として、前記データレジスタに格納された第 3 のビット幅の第 1 のデータ列に前記所定の演算を施して演算結果を第 3 のビット幅の第 2 のデータ列として前記データレジスタに再度格納させる第 2 の制御と、
 前記命令デコーダから与えられる信号に基づいて前記ア

ドレスレジスタに前記第1または第2のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記データレジスタに格納された前記第3のビット幅の第2のデータ列をバースト転送で前記第1または第2のメモリ領域に格納させる第3の制御と、前記命令デコーダから与えられる信号に基づいて前記第1、第2及び第3の制御を所定回数反復することにより、前記第2のデータ列を前記第1または第2のメモリ領域に連続的に格納する第4の制御とを実行すべくしてあることを特徴とするデータ処理装置。

【請求項3】 第1のビット幅のデータバスに接続された外部メモリと、命令をデコードする命令デコーダと、前記命令デコーダが命令をデコードした結果を示す信号が与えられることにより、命令の実行を制御するための制御信号を発生する実行制御部と、実行すべき命令の処理対象のデータを保持するデータレジスタと、前記処理対象のデータの前記外部メモリにおけるアドレスを保持するアドレスレジスタと、前記第1のビット幅より大きいまたは等しい第2のビット幅を処理単位として演算を施す少なくとも一つの演算器とを有し、前記実行制御部が出力する制御信号に従って命令を実行する命令実行部と、前記バスを通じて1回のアドレス出力に対して n (n は2以上の整数)回のデータ転送を行なうことにより、前記第2のビット幅より大きい第3のビット幅のデータをバースト転送して前記命令の実行に必要なデータを前記外部メモリとの間で入出力するバスインタフェイス部とを備えたデータ処理装置において、前記命令デコーダが、所定のビットパターンを前記外部メモリの任意のメモリ領域に反復して格納する多ビットデータ列演算命令をデコードした場合に、前記実行制御部は、前記命令デコーダから与えられる信号に基づいて、前記演算器に、前記第3のビット幅を処理単位として、前記所定のビットパターンを第3のビット幅のデータ列として前記データレジスタに格納させる第1の制御と、前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記任意のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記データレジスタに格納された前記第3のビット幅のデータ列をバースト転送で前記任意のメモリ領域に格納させる第2の制御と、前記命令デコーダから与えられる信号に基づいて前記第1及び第2の制御を所定回数反復することにより、前記第1のデータ列を前記任意のメモリ領域に連続的に格納する第3の制御とを実行し、前記命令デコーダが、前記第3の制御により前記外部メモリの任意の領域に格納されたデータを前記複数の内部レジスタに連続して格納する多ビットデータ列演算命令をデコードした場合に、前記実行制御部は、

3の制御とを実行すべくしてあることを特徴とするデータ処理装置。

【請求項4】 第1のビット幅のデータバスに接続された外部メモリと、命令をデコードする命令デコーダと、前記命令デコーダが命令をデコードした結果を示す信号が与えられることにより、命令の実行を制御するための制御信号を発生する実行制御部と、実行すべき命令の処理対象のデータを保持するデータレジスタと、前記処理対象のデータの前記外部メモリにおけるアドレスを保持するアドレスレジスタと、前記第1のビット幅より大きいまたは等しい第2のビット幅を処理単位として演算を施す少なくとも一つの演算器とを有し、前記実行制御部が出力する制御信号に従って命令を実行する命令実行部と、前記バスを通じて1回のアドレス出力に対して n (n は2以上の整数)回のデータ転送を行なうことにより、前記第2のビット幅より大きい第3のビット幅のデータをバースト転送して前記命令の実行に必要なデータを前記外部メモリとの間で入出力するバスインタフェイス部と、内部状態を保持するための複数の内部レジスタとを備えたデータ処理装置において、前記命令デコーダが、前記複数の内部レジスタが保持するデータを前記外部メモリの任意のメモリ領域に連続して格納する多ビットデータ列演算命令をデコードした場合に、前記実行制御部は、前記命令デコーダから与えられる信号に基づいて、前記演算器に、前記第3のビット幅を処理単位として、前記複数の内部レジスタが保持するデータを第3のビット幅のデータ列として前記データレジスタに格納させる第1の制御と、前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記任意のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記データレジスタに格納された前記第3のビット幅のデータ列をバースト転送で前記任意のメモリ領域に格納させる第2の制御と、前記命令デコーダから与えられる信号に基づいて前記第1及び第2の制御を所定回数反復することにより、前記第1のデータ列を前記任意のメモリ領域に連続的に格納する第3の制御とを実行し、前記命令デコーダが、前記第3の制御により前記外部メモリの任意の領域に格納されたデータを前記複数の内部レジスタに連続して格納する多ビットデータ列演算命令をデコードした場合に、前記実行制御部は、

前記命令デコーダから与えられる信号に基づいて前記アドレスレジスタに前記任意のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、前記命令実行部に、前記アドレスレジスタに設定されたアドレスに従って前記バスインタフェイス部を介して前記任意のメモリ領域に格納されている多ビットデータ列を前記第3のビット幅分だけ前記外部メモリからバースト転送で読み出して前記データレジスタに格納させる第4の制御と、

前記命令デコーダから与えられる信号に基づいて、前記演算器に、前記第3のビット幅を処理単位として、前記データレジスタに格納されたデータ列を前記複数の内部レジスタに格納させる第5の制御と、

前記命令デコーダから与えられる信号に基づいて前記第1及び第2の制御を所定回数反復することにより、前記データ列を前記複数の内部レジスタに連続的に格納する第6の制御とを実行すべくしてあることを特徴とするデータ処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明はストリング操作命令あるいはビットマップ操作命令を備え、更にそれらの命令を、CPUとメモリとの間でバースト転送機能を行なうことにより高速に処理するデータ処理装置に関する。

【0002】

【従来の技術】従来のデータ処理装置では8ビット、16ビット、または32ビットのデータエレメントが複数個連続したストリングの転送、比較、検索等の処理を行なう場合、エレメント単位の処理を複数回反復することにより、処理対象のストリング全体を処理していた。また、ワード長より長いビット列を操作する場合も、ワード長以下のビット列操作を複数回反復することによりビット列全体の操作を行っていた。エレメント単位のメモリアクセスを行なう命令あるいはワード長以下のビット列操作を行なう命令は個々の命令自体はメモリをランダムにアクセスする命令である。このため、そのような命令でメモリ中のデータを操作するということは、本来は連続したメモリ領域に格納されている連続したデータを小さなアクセス単位に分割してアクセスするという無駄な操作をしていることになる。

【0003】このような問題を解決するため、ストリングの操作を行なうストリング操作命令あるいはワード長より長いビット列を操作するビットマップ操作命令を有し、それらの命令をマイクロプログラムにより実行することにより、データバス幅単位でストリングあるいはビット列をアクセスして効率よく実行するデータ処理装置も提案されている。たとえば、ストリング操作命令を効率よく実行するデータ処理装置としては、特開平4-109337号公報に詳しく開示された発明が知られている。また、ビットマップ操作命令を効率よく実行するデータ処

理装置としては、特開昭64-88837号公報に詳しく開示された発明が知られている。

【0004】

【発明が解決しようとする課題】従来のデータ処理装置では、ストリング操作あるいはビット列操作を行なう場合、1回のバスサイクルで1ワードあるいはデータバス幅に一致するデータしか転送出来ないため、メモリとデータ処理装置とを結ぶバスの個々のデータの転送速度がデータ処理装置の性能に決定的な影響を与えていた。特に、メモリとしてダイナミックRAMチップ(DRAM)を用いた場合、CPUのバスアクセス性能に比してDRAMの動作速度が遅いため、バスアクセスに1乃至2ウェイト程度のウェイトサイクルを挿入する必要性が生じ、バスアクセス性能がデータ処理装置全体の性能を制限するという問題があった。

【0005】

【課題を解決するための手段】本発明のデータ処理装置は、第1、第2及び第3の発明では、第1のビット幅のデータバスに接続された外部メモリと、命令をデコードする命令デコーダと、命令デコーダが命令をデコードした結果を示す信号が与えられることにより、命令の実行を制御するための制御信号を発生する実行制御部と、実行すべき命令の処理対象のデータを保持するデータレジスタと、処理対象のデータの外部メモリにおけるアドレスを保持するアドレスレジスタと、第1のビット幅より大きいまたは等しい第2のビット幅を処理単位として演算を施す少なくとも一つの演算器とを有し、実行制御部が出力する制御信号に従って命令を実行する命令実行部と、バスを通じて1回のアドレス出力に対して n (n は2以上の整数)回のデータ転送を行なうことにより、第2のビット幅より大きい第3のビット幅のデータをバースト転送して命令の実行に必要なデータを外部メモリとの間で入出力するバスインタフェイス部とを備えている。

【0006】また、第4の発明では、上述の第1、第2及び第3の発明の構成に加えて、内部状態を保持するための複数の内部レジスタとを備えている。

【0007】

【作用】本発明のデータ処理装置の第1の発明では、命令デコーダが、外部メモリの第1のメモリ領域に連続して格納されている第1の多ビットデータ列と外部メモリの第2のメモリ領域に連続して格納されている第2の多ビットデータ列との各ビット間で所定の演算を順次施した結果の各ビットを第1または第2のメモリ領域に格納する多ビットデータ列演算命令をデコードした場合に、実行制御部は、命令デコーダから与えられる信号に基づいてアドレスレジスタに第1及び第2のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介して第1

及び第2のメモリ領域に格納されている第1及び第2の多ビットデータ列をそれぞれ第3のビット幅だけ外部メモリからバースト転送で読み出してデータレジスタに格納させる第1の制御と、命令デコーダから与えられる信号に基づいて所定の演算を指示する信号を与えることにより、演算器に、第2のビット幅を処理単位として、データレジスタに格納された第3のビット幅の第1のデータ列と第2のデータ列との間で所定の演算を施して演算結果を第3のビット幅の第3のデータ列としてデータレジスタに再度格納させる第2の制御と、命令デコーダから与えられる信号に基づいてアドレスレジスタに第1または第2のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介してデータレジスタに格納された第3のビット幅の第3のデータ列をバースト転送で第1または第2のメモリ領域に格納させる第3の制御と、命令デコーダから与えられる信号に基づいて第1、第2及び第3の制御を所定回数反復することにより、第3のデータ列を第1または第2のメモリ領域に連続的に格納する第4の制御とを実行する。

【0008】本発明のデータ処理装置の第2の発明では、命令デコーダが、外部メモリの第1のメモリ領域に連続して格納されている多ビットデータ列に所定の演算を順次施した結果の各ビットを第1のメモリ領域または第1のメモリ領域とは異なる第2のメモリ領域に格納する多ビットデータ列演算命令をデコードした場合に、実行制御部は、命令デコーダから与えられる信号に基づいてアドレスレジスタに第1のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介して第1のメモリ領域に格納されている多ビットデータ列を第3のビット幅分だけ外部メモリからバースト転送で読み出してデータレジスタに格納させる第1の制御と、命令デコーダから与えられる信号に基づいて所定の演算を指示する信号を与えることにより、演算器に、第3のビット幅を処理単位として、データレジスタに格納された第3のビット幅の第1のデータ列に所定の演算を施して演算結果を第3のビット幅の第2のデータ列としてデータレジスタに再度格納させる第2の制御と、命令デコーダから与えられる信号に基づいてアドレスレジスタに第1または第2のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介してデータレジスタに格納された第3のビット幅の第2のデータ列をバースト転送で第1または第2のメモリ領域に格納させる第3の制御と、命令デコーダから与えられる信号に基づいて第1、第2及び第3の制御を所定回数反復することにより、第2のデータ列を第1また

は第2のメモリ領域に連続的に格納する第4の制御とを実行する。

【0009】本発明のデータ処理装置の第3の発明では、命令デコーダが、所定のビットパターンを外部メモリの任意のメモリ領域に反復して格納する多ビットデータ列演算命令をデコードした場合に、実行制御部は、命令デコーダから与えられる信号に基づいて、演算器に、第3のビット幅を処理単位として、所定のビットパターンを第3のビット幅のデータ列としてデータレジスタに格納させる第1の制御と、命令デコーダから与えられる信号に基づいてアドレスレジスタに任意のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介してデータレジスタに格納された第3のビット幅のデータ列をバースト転送で任意のメモリ領域に格納させる第2の制御と、命令デコーダから与えられる信号に基づいて第1及び第2の制御を所定回数反復することにより、データ列を任意のメモリ領域に連続的に格納する第3の制御とを実行する。

【0010】本発明のデータ処理装置の第4の発明では、命令デコーダが、複数の内部レジスタが保持するデータを外部メモリの任意のメモリ領域に連続して格納する多ビットデータ列演算命令をデコードした場合に、実行制御部は、命令デコーダから与えられる信号に基づいて、演算器に、第3のビット幅を処理単位として、複数の内部レジスタが保持するデータを第3のビット幅のデータ列としてデータレジスタに格納させる第1の制御と、命令デコーダから与えられる信号に基づいてアドレスレジスタに任意のメモリ領域にデータを順次的に格納するためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介してデータレジスタに格納された第3のビット幅のデータ列をバースト転送で任意のメモリ領域に格納させる第2の制御と、命令デコーダから与えられる信号に基づいて第1及び第2の制御を所定回数反復することにより、第1のデータ列を任意のメモリ領域に連続的に格納する第3の制御とを実行し、命令デコーダが、第3の制御により外部メモリの任意の領域に格納されたデータを複数の内部レジスタに連続して格納する多ビットデータ列演算命令をデコードした場合に、実行制御部は、命令デコーダから与えられる信号に基づいてアドレスレジスタに任意のメモリ領域からデータを順次的に読み出すためのアドレスを設定することにより、命令実行部に、アドレスレジスタに設定されたアドレスに従ってバスインタフェイス部を介して任意のメモリ領域に格納されている多ビットデータ列を第3のビット幅分だけ外部メモリからバースト転送で読み出してデータレジスタに格納させる第4の制御と、命令デコーダから与えられる信号に基づいて、演算器に、第3のビ

ット幅を処理単位として、データレジスタに格納されたデータ列を複数の内部レジスタに格納させる第5の制御と、命令デコーダから与えられる信号に基づいて第1及び第2の制御を所定回数反復することにより、データ列を複数の内部レジスタに連続的に格納する第6の制御とを実行する。

【0011】

【実施例】以下、本発明をその実施例を示す図面に基づいて詳述する。

【0012】(1)「本発明のデータ処理装置を用いたシステムの構成」

図1のブロック図に本発明のデータ処理装置を使用したシステム構成例を示す。なお、図1においては、参照符号100が本発明のデータ処理装置を示している。本発明のデータ処理装置100は32ビット幅のアドレスバス101及び32ビット幅のデータバス102とを介して主記憶装置103及び周辺回路105と接続されていてそれらをアクセスする。各ユーザが固有に必要とする制御回路、DMAC、タイマ等は周辺回路105に内蔵されていることを前提とし、本発明のデータ処理装置100にはそれらの回路は内蔵されていない。

【0013】なお、周辺回路105は上述のような各ユーザが固有に必要とする制御回路、DMAC、タイマ等を内蔵したASICとして構成することが出来る。また、主記憶装置103はDRAMあるいはPROMで構成することが出来る。主記憶装置103へのアクセスは、1回のバスサイクルにアドレスを1つ出力して32ビット以下のデータをアクセスする単一転送と、1回のバスサイクルにアドレスを1つ出力して4つの32ビットデータ(全体で16バイト)をアクセスするバースト転送とのいずれかが可能である。

【0014】図2には主記憶装置103及びその制御のための構成の一例を示すブロック図である。なお、ここでは一例として、主記憶装置103としてCDRAM(Cache-DRAM)を使用したメモリシステムの例を示す。

【0015】参照符号111はメモリコントローラであり、周辺回路105に内蔵されている。この例ではCDRAM110としては三菱電機株式会社製のM5M44409を8個使用するとにより、16ワード×256エントリのダイレクトマップ方式キャッシュ付き4Mバイトメモリを実現している。なお、M5M44409はそれぞれが16KバイトのSRAM110Sと4MバイトのDRAM110Dとを内蔵し、SRAM110Sがブロックサイズ=16ワードのキャッシュメモリとして動作する4Mバイトの同期式DRAMである。CDRAM110ではSRAM110SがDRAM110Dへのライトアクセスに対してストアバッファとして動作するため、SRAM110Sがヒットした場合はライトアクセスもリードアクセスと同一タイミングで高速に行なうことができる。

【0016】CDRAM110がヒットした場合、本発明のデータ処理装置100はゼロウェイットで単一転送またはバーストモードによるブロック転送ができる。CDRAM110が

ミスした場合でも第2～4転送をゼロウェイットのバーストモードによるブロック転送で実行することが可能である。バーストモードによるブロック転送を行なう場合、メモリコントローラ111は#HIT信号をアサートして、アドレスのビット28, 29を昇順にラップアラウンドしてCDRAM110をアクセスする。

【0017】図3の模式図にCDRAM110を用いたメモリシステムのアドレス割り付け方法を示す。アドレスのビット0～9(A0:9)は4Mバイトのメモリ空間を選択するためのチップセレクトアドレスである。キャッシュアクセスに際しては、ビット10～17(A10:17)はキャッシュのタグアドレスとして使用される。ビット18～25(A18:25)はエントリアドレスの指定に使用される。ビット26～29(A26:29)はブロック内のワードアドレスの指定に使用される。また、DRAMアクセスに際しては、ビット10～19(A10:19)はロウアドレス、ビット20～25(A20:25)はカラムアドレスとして使用される。

【0018】以下、本発明のデータ処理装置100の命令体系、処理機構及びバスオペレーション機構について最初に説明し、次にストリング操作命令、ビットマップ操作命令の詳細動作について説明する。

【0019】(2)「本発明のデータ処理装置の命令」

(2.1)「命令フォーマット」

本発明のデータ処理装置100の命令は16ビット単位で可変長となっており、奇数バイト長の命令はない。本発明のデータ処理装置100では高頻度に使用される命令を短いフォーマットとするため、特に工夫された命令フォーマット体系を有する。たとえば、2オペランド命令に対しては基本的に「4バイト+拡張部」の構成を有し、全てのアドレッシングモードが利用可能な一般形フォーマットと、頻度の高い命令とアドレッシングモードとのみを使用可能な短縮形フォーマットとの2つのフォーマットを有する。図4、図5、図6及び図7は本発明のデータ処理装置100の命令フォーマットを示す模式図である。各図中に現われる記号の意味は以下の通りである。

【0020】—:オペレーションコードの入る部分

Ea: 8ビットの一般形のアドレッシングモードでオペランドを指定する部分

Sh: 6ビットの短縮形のアドレッシングモードでオペランドを指定する部分

Rn: レジスタファイル上のオペランドをレジスタ番号で指定する部分

【0021】フォーマットは、図4に示すように右側がLSB側で、かつ高いアドレスになっている。アドレスNとアドレスN+1との2バイトを見た後でないと命令フォーマットが判別できないようになっているが、これは命令が必ず16ビット(ハーフワード)単位でフェッチ、デコードされることを前提としたためである。

【0022】本発明のデータ処理装置100の命令では、いずれのフォーマットの場合も、各オペランドのEaまた

はShの拡張部は必ずそのEaまたはShの基本部を含む16ビット(ハーフワード)の直後に置かれる。これは、命令により暗黙に指定される即値データ及び命令の拡張部に優先する。従って、4バイト以上の命令では、Eaの拡張部によって命令のオペレーションコードが分断される場合がある。なお、本発明のデータ処理装置100の命令フォーマットに関しては特開昭64-91228号公報、U.S.P. No. 5,029,069等に詳細に開示されている。

【0023】(2.2)「アドレッシングモード」

本発明のデータ処理装置100の命令のアドレッシングモード指定方法には、レジスタを含めて6ビットで指定する短縮形と、8ビットで指定する一般形とがある。未定義のアドレッシングモードを指定した場合、あるいは意味的に考えて明らかに不合理なアドレッシングモードの組み合わせが指定された場合には、未定義命令を実行した場合と同様に予約命令例外が発生されて例外処理が起動される。これに該当するのは、デスティネーションが即値モードである場合、アドレス計算を伴うべきアドレッシングモード指定フィールドで即値モードが使用された場合などである。

【0024】本発明のデータ処理装置100の命令は種々のアドレッシングモードをサポートする。本発明のデータ処理装置100でサポートするアドレッシングモードには、レジスタ直接モード、レジスタ間接モード、レジスタ相対間接モード、即値モード、絶対モード、PC相対間接モード、スタックポップモード、スタックプッシュモード及び多段間接モードがある。レジスタ直接モードは、レジスタの内容をそのままオペランドとするアドレッシングモードである。レジスタ間接モードは、汎用レジスタの内容をアドレスとするメモリの内容をオペランドとするアドレッシングモードである。レジスタ相対間接モードは、ディスプレイメント値が16ビットであるか32ビットであるかにより2種類に分かれる。いずれも、汎用レジスタの内容に16ビットまたは32ビットのディスプレイメント値を加えた値をアドレスとするメモリの内容をオペランドとするアドレッシングモードである。ディスプレイメント値は符号付きとして扱われる。

【0025】即値モードは、命令コード中で指定されるビットパターンをそのまま2進数と見なしオペランドとするアドレッシングモードである。即値のサイズはオペランドサイズとして命令中で指定される。絶対モードは、アドレス値が16ビットで示されるか32ビットで示されるかにより2種類に分かれる。いずれも、命令コード中で指定される16ビットまたは32ビットのビットパターンをアドレスとしたメモリの内容をオペランドとするアドレッシングモードである。16ビットでアドレスが示される場合は指定されたアドレス値を32ビットに符号拡張される。PC相対間接モードは、ディスプレイメント値が16ビットであるか32ビットかにより2種類に分かれ

る。いずれも、プログラムカウンタの内容に16ビットまたは32ビットのディスプレイメント値を加えた値をアドレスとするメモリの内容をオペランドとするアドレッシングモードである。ディスプレイメント値は符号付きとして扱われる。PC相対間接モードにおいて参照されるプログラムカウンタの値は、そのオペランドを含む命令の先頭アドレスである。

【0026】スタックポップモードは、スタックポイント(SP)の内容をアドレスとするメモリの内容をオペランドとするアドレッシングモードである。オペランドアクセス後にSPをオペランドサイズだけインクリメントする。たとえば、32ビットデータが扱われる場合には、オペランドアクセス後にSPが+4だけ更新される。8, 16, 64ビットのサイズのオペランドに対するスタックポップモードの指定も可能であり、それぞれSPが+1, +2, +8だけ更新される。スタックプッシュモードは、SPの内容をオペランドサイズだけデクリメントした内容をアドレスとするメモリの内容をオペランドとするアドレッシングモードである。スタックプッシュモードではオペランドアクセス前にSPがデクリメントされる。たとえば、32ビットデータが扱われる場合には、オペランドアクセス前にSPが-4だけ更新される。8, 16, 64ビットのサイズのオペランドに対するスタックプッシュモードの指定も可能であり、それぞれSPが-1, -2, -8だけ更新される。

【0027】多段間接モードは、汎用レジスタあるいはPCの値をベースアドレスとし、その値にインデックスレジスタ値あるいはディスプレイメント値を加算した値をアドレスとしたり、あるいはそのアドレスを基にメモリ間接アドレッシングをするアドレッシングモードである。なお、本発明のデータ処理装置100の命令のアドレッシングモードに関して本発明の命令フォーマットと同様に特開昭64-91253号公報に詳しく記載されている。

【0028】(2.3)「ストリング操作命令」

ストリングとは、8ビット、16ビット、または32ビットのデータエレメントを任意の長さだけ連続して並べたデータタイプである。個々のデータの内容には制限はなく、実際の文字コードになる場合、整数になる場合、浮動小数点数になる場合などがあり、ユーザ側で解釈する。ストリングのサイズは命令オプションの一部として指定され、本発明のデータ処理装置100のストリング操作命令で指定可能なエレメントのサイズ(sx)とそのビットパターンとは以下ようになる。

【0029】sx=00 1バイト

sx=01 2バイト

sx=11 4バイト

sx=11 指定できない

【0030】ストリングの範囲を示す方法には、ストリングの長さ(データ数)を指定する方法と、ストリングの終了を示す文字(ターミネータ)を指定する方法との

2通りがあり、使用目的に応じて適当に選択すればよい。本発明のデータ処理装置100のストリング操作命令ではストリングの個数がパラメータとなっているが、更に、割り出し条件という形でターミネータあるいは命令の終了条件を与えることも可能であり、両方の指定方法を利用することができる。割り出し条件の指定は、命令オプションの一部として行なわれる。

【0031】本発明のデータ処理装置100では、ストリング操作命令SMOV, SCMP, SSCHの割り出し条件として大小比較あるいは二値比較を含む豊富な条件が指定可能であり、大きな特徴となっている。特に、ストリングサーチ用のSSCH命令は、検索条件が割り出し条件として指定されるため、割り出し条件にのみ意味がある命令となっている。本発明のデータ処理装置100のストリング操作命令で指定可能な条件(eeee)とそのビットパターンとは図100に一覧表として示すようになる。

【0032】また、ストリングサーチ命令(SSCH命令)では、連続領域に置かれたデータの集合のみならず、一定間隔で飛び飛びのアドレスに置かれたデータの集合もストリングとしてサポートしている。即ち、この命令に限ってはポインタの増減量を自由に設定可能であり、テーブル検索あるいは多次元配列のスキャン等にこの命令を利用することができる。ところで、ストリング操作命令の用途としては、文字どおり8/16ビットの文字列を処理する場合の他に、たとえば特定のビットパターンのサーチ、メモリのブロック転送、構造体の代入、メモリ領域のクリア等への応用が可能である。

【0033】ストリング操作命令は後述する任意長ビットフィールド命令と同様に不定長のデータを扱うため、実行中の割り込み受け付け及び実行の再開の機能が不可欠である。一方、ストリング操作命令自体がコンパイラが生成するコードとなる可能性はほとんどなく、アセンブラで記述されたサブルーチンとして提供される場合が多い。このため、対称性あるいはアドレッシングモードについての制限はあまり問題にならない。従って、本発明のデータ処理装置100におけるストリング操作命令では、オペランドあるいは実行途中の状態保持のために、内蔵する固定番号のレジスタ(R0~R4)を使うようになっている。主なレジスタの使い方は次のようになる。

【0034】R0: ソース側ストリング(src)の先頭アドレス

R1: デスティネーション側ストリング(dest)の先頭アドレス

R2: ストリングの長さ、エレメント数

R3: 割り出し条件の比較値

【0035】これらの内のストリングの長さを表わすR2はエレメント数であって、バイト数ではない。R2は符号なしの数として扱われ、R2=0の場合はエレメント数による命令終了は行なわないという意味に解釈される。つまり、エレメント数による終了を避けたい場合には、R2

=0として命令を実行すればよいことになる。SMOV命令はストリングの転送を行なう命令である。命令のビットパターンと各種オプション機能とを図8に示す。

【0036】SCMP命令は2つのストリングの比較を行なう命令である。命令のビットパターンと各種オプション機能とを図9に示す。SSCH命令はストリングのサーチを行なう命令である。命令のビットパターンと各種オプション機能とを図10に示す。SSTR命令はR3レジスタで指定したエレメントを複数個メモリにストアする命令である。命令のビットパターンと各種オプション機能とを図11に示す。

【0037】(2.4)「ビットマップ操作命令」

本発明のデータ処理装置100はビットマップディスプレイの操作を行なう場合等に有効な命令として、ワード長を超える任意長のビット列を操作するビットマップ操作命令を備える。ビットマップ操作命令には、ビット列の一般的な演算と転送を行なうBVMAP命令、ビット列の転送を行なうBVCPY命令、繰り返しパターンの演算と転送を行なうBVPAT命令、ビット列中の"0"または"1"のビットをサーチするBVSCH命令がある。これらの内、BVMAP, BVPAT, BVCPY命令は特にビットマップディスプレイ上のウインドウ操作(bitblt)を主な目的とした命令である。

【0038】ビットマップ操作命令はオペランドが多く、且つ実行時間も長い。従って、命令実行中での割り込みの受け付けのメカニズム及び割り込み処理後の再実行のメカニズムが必要になる。本発明のデータ処理装置100では、オペランドの指定と演算の進行状況の表現とのために固定番号のレジスタを使用している。そのため、ビットマップ操作命令実行中に割り込みが入っても、割り込み処理ハンドラ中でそのレジスタの退避と復帰とが正しく行なわれていれば、割り込み処理後にそのビットマップ操作命令を途中から再開することが可能である。従って、実行中断後に状態の退避あるいはコンテキストスイッチを行なったり、コンテキストスイッチ後に別のコンテキストで同じビットマップ操作を実行し、再び前のコンテキストに戻って前のビットマップ操作命令を再開したとしても、問題なく動作する。

【0039】BVSCH命令は任意長ビット列中から"0"または"1"のビットをサーチする命令である。命令のビットパターンと各種オプション機能及びレジスタ上のパラメータとを図12に示す。BVMAP命令は任意長ビット列の演算を行なう命令である。命令のビットパターンと各種オプション機能及びレジスタ上のパラメータとを図13に示す。BVCPY命令は任意長ビット列のコピーを行なう命令である。命令のビットパターンと各種オプション機能及びレジスタ上のパラメータとを図14に示す。BVPAT命令は任意長ビット列とパターンとの演算を行なう命令である。命令のビットパターンと各種オプション機能及びレジスタ上のパラメータとを図15に示す。

【0040】(3)「本発明のデータ処理装置の機能ブロック」

(3.1)「機能ブロックの構成」

図16に本発明のデータ処理装置100のブロック図を示す。本発明のデータ処理装置100の内部を機能的に大きく分けると、命令フェッチ部(IFU)150、命令デコード部(DU)151、ROM部(RU)152、アドレス生成部(AGU)153、PC生成部(PCU)154、整数演算部(IU)155、オペランドアクセス部(OAU)156、バスインターフェイス部(BIU)157に分かれる。バスインターフェイス部157は、外部とはアドレスバス101(A00:31)、データバス102(D00:31)及び各種制御信号により結合している。

【0041】命令フェッチ部150は命令を内蔵命令キャッシュ200(図18参照)あるいは外部のメモリ(主記憶装置103)からフェッチする。命令デコード部151は命令フェッチ部150からIバス169で転送された命令をデコードする。ROM部152はマイクロプログラムに従って整数演算部155を制御する。PC生成部154は命令のPC値を計算する。アドレス生成部153はオペランドのアドレスを計算する。オペランドアクセス部156は内蔵データキャッシュ200あるいは外部のメモリからオペランドをフェッチしたり、外部のメモリへオペランドをストアする処理を行なう。

【0042】命令はバスインターフェイス部157から命令フェッチ部150に取り込まれ、命令デコード部151でデコードされ、ROM部152でのマイクロプログラム制御により整数演算部155で実行される。命令のPC値の計算、オペランドのアドレス計算、オペランドのアクセスは整数演算とは独立のブロックでハードワイヤード制御により行なわれる。命令のアドレスはJAバス171で各部から命令フェッチ部150へ転送される。オペランドアドレスはAAバス160で各部からオペランドフェッチ部へ転送される。整数演算部155とオペランドアクセス部156との間のオペランドの転送はDDバス164により行なわれる。

【0043】(3.2)「バスインターフェイス部」

バスインターフェイス部157は命令フェッチ部150あるいはオペランドアクセス部156の要求に従ってバスサイクルを発行し、外部のメモリ、即ち主記憶装置103にアクセスする。本発明のデータ処理装置100におけるメモリアクセスはクロック同期のプロトコルで行なわれる。1回のバスサイクルには最小2クロックサイクルが必要である。リードサイクルには1回のバス転送で4バイト境界内の命令コードあるいはデータをフェッチする単一転送と、4回のバス転送で16バイト境界内の命令コードあるいはデータを一度にフェッチするブロック転送とがある。更に、ブロック転送にはアドレスを1回出力して、命令コードあるいはデータを4回連続フェッチするバーストモードとアドレスとを4回出力して、命令コードあるいはデータを4回連続フェッチするクワッドムー

ブモードがある。

【0044】ライトサイクルについてもリードサイクルと同様に1回のバス転送で4バイト境界内のデータをストアする単一転送と、4回のバス転送で16バイト境界内のデータを一度にストアするブロック転送とがある。ライトサイクルのブロック転送にもアドレスを1回出力してデータを4回連続ストアするバーストモードと、アドレスを4回出力してデータを4回連続ストアするクワッドムーブモードとがある。各バスサイクルの制御、各種信号の入出力はバスインターフェイスでハードワイヤード制御により行なわれる。

【0045】バスインターフェイス部157のブロック図を図17に示す。なお、図17には命令フェッチ部150及びオペランドアクセス部156も共に示されている。オペランドアクセス部156、命令フェッチ部150とアドレスパッド180(A-pad)とはBAバス162で結合される。RDADDR182は命令フェッチまたはデータリード時にIAバス167あるいはOAバス172から転送されてきたアドレスをBAバス162へ出力するレジスタである。

【0046】SBADDR183はストアバッファ292のアドレス保持部(レジスタ)であり、2つのアドレスを保持する。BA184はデータまたは命令のブロック転送時にアドレスをインクリメントする回路である。バスエラーが発生した場合はそのアドレスと入出力情報とがバスインターフェイス157のバスエラー情報レジスタBERRINF185に保持され、S1バス10へ出力される。オペランドアクセス部156及び命令フェッチ部150はデータパッド181(D-pad)とはBDバス163で結合されている。

【0047】データリードに際しては、オペランドアクセス部156からOAバス172、RDADDR182、BAバス162を通じて外部のアドレスバス101へ出力されたアドレスに従って、バスインターフェイス部157がデータバス102からデータをフェッチし、BDバス163を通じてオペランドアクセス部156へ転送する。ブロック転送のクワッドムーブモードでデータをフェッチする際は、オペランドアクセス部156からは1つのアドレスが出力され、残り3つのアドレスはBAレジスタが下位ビットをラップアラウンドして出力する。

【0048】データライトに際しては、オペランドアクセス部156からOAバス172、SBADDR183、BAバス162を通じてアドレスバス101へアドレスが出力され、BDバス163を通じてデータバス102へデータが出力される。ブロック転送のクワッドムーブモードでデータをライトする際は、オペランドアクセス部156からは1つのアドレスが出力され、残り3つのアドレスはBAレジスタが下位ビットをラップアラウンドして出力する。

【0049】命令フェッチに際しては、命令フェッチ部150からIAバス167、RDADDR182、BAバス162を通じて外部のアドレスバス101へ出力されたアドレスに従って、バスインターフェイス部157がデータバス102から命令

コードをフェッチし、BDバス163 を通じて命令フェッチ部150 へ転送する。ブロック転送のクワッドムーブモードで命令コードをフェッチする際は、命令フェッチ部150 からは1つのアドレスが出力され、残り3つのアドレスはBAレジスタが下位ビットをラップアラウンドし出力する。

【0050】バスインターフェイス部157 はメモリアクセスの他、外部割り込みの受け付け、バスアービトリションの制御も行なう。本発明のデータ処理装置100 以外の外部デバイスがバスマスタになっており本発明のデータ処理装置100 がバスマスタ動作中である場合は、外部デバイスがデータライトまたは無効化サイクルを実行した場合にアドレスバス101 上に出力されたアドレスをバスインターフェイス部157 が取り込んでBAバス162 を通じて命令フェッチ部150 とオペランドアクセス部156 とへ転送する。バスマスタ動作でのアドレスの取り込み動作は、バス権開放中（後述する #HACK信号609 がアサート中）に #DS信号604 がアサートされた場合にクロック非同期に行なわれる。

【0051】(3.3) 「命令フェッチ部」

命令フェッチ部150 のブロック図を図18に示す。なお、図18には命令デコード部151 及びバスインターフェイス部157 も共に示されている。命令フェッチ部150 には4KBの内蔵命令キャッシュ200、共に16バイトの命令キュー(A, B)201 が2つとそれらの制御部とが備えられている。命令フェッチ部150 は、次にフェッチすべき命令のアドレスに従って内蔵命令キャッシュ200 から命令コードをフェッチして命令キュー201 を介して命令デコード部151 へ転送する。バスインターフェイス部157 と命令キャッシュ200 との間は32ビットのアドレスバス101(IAバス167)と32ビットのデータバス102(BDバス163)とで結合されている。命令キャッシュ200 と命令キュー201 との間は64ビットの ICOUTバス168 で結合されている。

【0052】バスインターフェイス部157 から出力された命令はBDバス163 を通じて命令キャッシュ200 へ転送される。命令キュー201 から出力された命令コードはIIバス169 を通じて命令デコード部151 へ転送される。命令のアドレスはJAバス171 からIAレジスタ203 を経由してアドレスデコード機構204 と命令キャッシュ200 とに入力される。

【0053】命令キャッシュ200 は16byte×256entry構成のダイレクトマップ制御で動作する。命令キャッシュ200 はIAレジスタ203 から転送されたアドレスの下位12ビットに従ってキャッシュアドレスタグと命令コードとを出力する。キャッシュアドレスタグはアドレスの上位20ビットと比較され、一致すれば命令キャッシュ200はヒットとなり、ICOUTバス168 を経由して命令コードが命令キュー201 へ転送される。この際、命令キュー201 が空である場合には、命令コードは命令キュー201 をバイパスして同一タイミングでIIバス169 を通じて命令デ

コード部151へ転送される。命令キャッシュ200 がミスした場合はアドレスデコード機構204からIAバス167 を通じてバスインターフェイス部157 へ命令アドレスが出力され、外部のメモリ、即ち主記憶装置103 がブロック転送でアクセスされて命令コードがフェッチされ、命令キャッシュ200 のエントリが更新される。

【0054】アドレスデコード機構204 にはレジスタI0 ADDR, IOMASK, NCADDRがあり、命令アドレスが I/O領域に入るか否か、非キャッシュ領域に入るか否かのチェックを行なう。また、アドレスデコード機構204 ではOAバス172 から転送されてきたオペランドアドレスに関してもそのアドレスが I/O領域に入るか否か、非キャッシュ領域に入るか否かのチェックを行なう。

【0055】2つの命令キュー201 の内の1つは条件分岐命令に連続する命令コードをプリフェッチしてキューイングし、もう1つは条件分岐命令の分岐先の命令コードをプリフェッチしてキューイングする。2つの命令キュー201 が共に命令をフェッチした状態で更に条件分岐命令をデコードした場合の分岐先アドレスは分岐先アドレスレジスタ(BAB)202に保持され、先行する条件分岐命令が実行されどちらかの命令キュー201 がクリアされた後にこのアドレスから新たに命令がフェッチされる。ジャンプ先以外の命令アドレスは各命令キュー201 の専用カウンタで計算される。ジャンプが発生した場合は、新たな命令アドレスが、アドレス生成部153、PC生成部154あるいは整数演算部155 からJAバス171 により各命令キュー201のカウンタにロードされる。

【0056】また、本発明のデータ処理装置100 がバスマスタ動作中である場合はバスインターフェイス部157 が外部のアドレスバス101 上のアドレスをモニタし、その結果がBAバス162 を通じて命令キャッシュ200 へ転送される。バスインターフェイス部157 は本発明のデータ処理装置100 自身がライト動作を行なったメモリのアドレスも命令キャッシュ200 へ転送する。

【0057】(3.4) 「命令デコード部」

本発明のデータ処理装置100 における命令は図19の模式図に示すように2バイト単位の変長命令であり、基本的には「2バイトの命令基本部+0~4バイトのアドレッシング修飾部」を1~3回反復することにより命令が構成されている。命令基本部にはオペレーションコード部とアドレッシングモード指定部とがあり、インデックスアドレッシングあるいはメモリ間接アドレッシングが必要な場合にはアドレッシング修飾部の代わりに「2バイトの多段間接モード指定部+0~4バイトのアドレッシング修飾部」が必要な数だけ拡張される。また、命令によっては2または4バイトの命令固有の拡張部が最後に付く場合もある。

【0058】本発明のデータ処理装置100 では図19に示した変長フォーマットの命令を効率よく処理するため、命令デコード段階で1つの命令を1つまたは複数の

10

20

30

40

50

処理単位（ステップコード）に分解する。基本的には「2 バイトの命令基本部+0~4 バイトのアドレッシング修飾部」または「多段間接モード指定部+アドレッシング修飾部」のデコード結果から1つのステップコードが生成され、命令デコーダは1クロックに1つのステップコードを出力する。しかし、第1オペランドにアドレッシング修飾部がない場合、または先頭の命令基本部にオペランド指定子がない命令では1クロックに2つの命令基本部（32ビット長）がデコードされる。また、本発明のデータ処理装置100では一部のレジスタ間演算命令を先行する命令と並列にデコードし、2命令を同時にデコードすることも可能である。

【0059】本発明のデータ処理装置100の命令デコード部151のブロック図を図20に示す。なお、図20にはROM部152、アドレス生成部153及びPC生成部154も共に示されている。命令デコード部151は、命令デコードステージ401(Dステージ)とこのDステージ401に引き続くオペランドアドレス生成ステージ402(Aステージ)との2つのパイプラインステージで動作する（なお、パイプラインステージに関しては詳しくは図27を参照して後述する）。命令デコード部151の構成要素の内Dステージ401で動作するのは、分岐予測部210、メインデコーダ211、第1サブデコーダ212、第2サブデコーダ213、アドレッシングモードデコーダ214及び拡張データ処理部215である。また、命令デコード部151の構成要素の内Aステージ402で動作する要素は、Aステージデコーダ216及びサブコード転送部217である。

【0060】メインデコーダ211は最大4バイトの命令基本部と1ビットの分岐予測ビットとをデコードする。メインデコーダ211の出力(Dコード222)はAステージデコーダ216へ転送される。分岐予測部210は1ビット×1Kエントリの条件分岐命令の分岐履歴を保持しており、条件分岐命令の直前にデコードした命令のPC値の下位アドレスに従って分岐予測ビットを出力する。

【0061】第1サブデコーダ212と第2サブデコーダ213とはメインデコーダ211でデコードされる命令に引き続く16ビットのレジスタ間演算命令をデコードする。第1サブデコーダ212はメインデコーダ211でデコードされる命令の命令長を16ビットと仮定してIIバス169の3バイト目と4バイト目とをデコードする。第2サブデコーダ213はメインデコーダ211でデコードされる命令の命令長を32ビットと仮定してIIバス169の5バイト目と6バイト目とをデコードする。メインデコーダ211でデコードされた命令とサブデコーダでデコードされた命令とが後述する並列デコード条件を満足する場合、2つのサブデコーダ212、213の出力の内のいずれか一方がマルチプレクサ(MUX)218により選択されてサブコード転送部217へ転送される。なお、マルチプレクサ218による選択はメインデコーダ211により制御される。

【0062】アドレッシングモードデコーダ214は8ビ

ットのオペランド指定子あるいは16ビットの多段間接モードフィールドで指定される命令のアドレッシングモード部をデコードしてアドレス生成部153を制御するAコード220を出力する。拡張データ処理部215はディスペースメント、即値等のアドレッシング修飾部を命令コードから取り出し、DISPバス165を通じてアドレス生成部153及びPC生成部154へ転送する。Aステージデコーダ216はメインデコーダ211から出力される命令の中間デコード結果であるDコード222を更に詳細にデコードし、マイクロプログラムのエントリ番地及びパラメータをROM部152へ出力する。Aステージデコーダ216ではLDM、STM等のような複数のオペランドをレジスタとメモリとの間で転送する命令を1度に8バイト以下のデータを転送する複数のメモリーレジスタ間転送命令(ステップコード)に分解する。この際、Aステージデコーダ216はアドレス生成部153にAAコード221を出力し、分解したオペランドのアドレス計算を制御する。

【0063】サブコード転送部217はいずれかのサブデコーダ212または213から出力されたデコード結果をAステージデコーダ216でデコードされる命令と同期させてROM部152へ転送する。また、Aステージデコーダ216とサブコード転送部217とはオペランドを書き込むレジスタの番号がアドレス生成部153のスコアボードレジスタ480(図54参照)へ転送され、パイプラインインターロック機構により書き込みが終了していないレジスタ値を後続命令がオペランドアドレス計算に使用してRAWデータハザード(Read-after-write data hazard)を起こさないように制御する。

【0064】(3.5)「PC生成部」

PC生成部154のブロック図を図21に示す。なお、図21には命令デコード部151も共に示されている。PC生成部154は、命令デコードステージ401での先行ジャンプ先アドレスを計算するジャンプ先アドレス生成部240、デコードした命令のPC値を計算するPC計算部241、パイプライン中で処理される命令の流れに同期してPC値を転送するPC転送部242及びサブルーチンからのプリリターン先アドレスを管理するPCスタック243を備えている。

【0065】ジャンプ先アドレス生成部240は命令デコードステージ401でPC相対ジャンプまたは絶対アドレスジャンプが行なわれた場合にジャンプ先命令のアドレスを計算する。ジャンプ先アドレス生成部240にはIIバス169とDPCバス166とに結合した2つの加算器(JTADD1, JTADD2)243、244が備えられており、IIバス169の分岐変位フィールドとなり得る3種類のフィールド(24:31, 16:31, 16:47)とビット8:15の内の一つとDPCバス166から転送されたデコード開始命令アドレスとを命令のデコードと並行して加算する。

【0066】また、符号拡張器(ABS)247はIIバス169から転送された絶対アドレスとなり得る3つのフィールド(24:31, 16:31, 16:47)の符号拡張を行なう。デコード

した命令がPC相対ジャンプを行なう命令(BRA, BSR, Bcc, JMP @ (disp:32, PC), JSR@ (disp:32, PC)) または絶対アドレスジャンプを行なう命令(JMP @abs, JSR @abs)である場合、2つの加算器243, 244での加算結果と1つの符号拡張器247での符号拡張結果とのいずれかから正しいジャンプ先アドレス1つが選択されてJAバス171へ出力される。また、命令のデコードと同時に求められたジャンプ先アドレス以外に分岐する一部のジャンプ命令(ACB, SCBの一部)に対しては、デコーダのハードウェアの都合により命令をデコードした後にJTADD2 244で新たにDISPバス165から転送される分岐変位とPC値とを加算してジャンプ先アドレスを求め、JAバス171へ出力する。

【0067】PCスタック243はサブルーチンジャンプ命令(BSR, JSR)からのリターン先アドレスのコピーを16エントリのスタックに保持しており、サブルーチンリターン命令(RTS, EXITD)がデコードされた場合にリターン先アドレスをJAバス171へ出力する。また、サブルーチンジャンプ命令が実行された場合、PCスタック243にはリターン先アドレスがS1バス10から転送されてプッシュされる。タスクスイッチによりスタックが切り替えられたり、16レベル以上のサブルーチンのネスティング等が発生した場合には、PCスタック243からJAバス171へ出力されるリターン先アドレスは正しいリターン先アドレスではなくなる。このため、サブルーチンリターン命令がパイプラインの実行ステージ404に到達した時点で、プリリターンアドレスがPCスタック243から再びS1バス10へ読み出され、メモリから読み出された正しいリターン先アドレスと比較される。

【0068】PC計算部241はPC加算器248と作業用レジスタ(TPC, ANPC, DPC)249とで構成されており、命令デコード部151でデコードする命令のPC値を計算する。このPC計算部241でのPC値の計算は、1サイクル前にデコードした命令のアドレスに命令デコード部151で消費した命令コード長を加算することにより行なわれる。ジャンプ命令あるいはEITにより命令実行のシーケンスが変更された場合は、JAバス171からPC計算部241へジャンプ先命令のアドレスが転送される。PC計算部241での計算結果は命令のデコード結果と共にパイプラインの流れに同期してPC転送部242へ出力される他、PC相対アドレス計算のためにIXバス170からアドレス生成部153へ転送され、また命令アドレスの計算のためにDPCバス166からジャンプ先アドレス生成部240へも転送される。

【0069】PC転送部242には本発明のデータ処理装置100の各パイプラインステージに対応する多数のPC保持用レジスタが備えられている。PC計算部241で計算された命令の先頭アドレスであるPC値は本発明のデータ処理装置100の各パイプラインステージで処理されるステップコードと共にPC転送部242のレジスタ中を転送される。PC転送部242ではデバッグサポートのための命令ア

ドレスブレイクまたはトレース動作の起動のために転送されるPC値と命令ブレイクポイントアドレス(IBA0, IBA1)の値と、トリガ開始命令アドレス(TGIA)の値との比較動作も行なう。

【0070】(3.6) 「アドレス生成部」

アドレス生成部153は、命令デコード部151のアドレッシングモードデコーダ214またはAステージデコーダ216から出力されたオペランドのアドレス生成に関係する制御情報によりハードワイヤード制御され、オペランドのアドレスを生成する。また、命令デコード部151で先行ジャンプを行なわないレジスタ間接アドレッシングのジャンプ命令による先行ジャンプ処理、条件分岐命令の分岐予測と反対側の命令アドレスの計算あるいはサブルーチンジャンプ命令の戻り先アドレスの計算も行なう。

【0071】図22にアドレス生成部153のブロック図を示す。アドレス生成部153は大きくはSP先行更新部260とアドレス計算部261で構成されている。SP先行更新部260は、スタックポップアドレッシング(@SP+)あるいはスタックプッシュアドレッシング(@-SP)が連続してもパイプラインインターロックなしに命令を処理するためのオペランドアドレス生成専用のSP(ASP)262と、オペランドのアドレス生成段階で更新したSPの値を命令の流れに同期してパイプライン中で転送するレジスタ群(SP転送部)270とで構成される。また、アドレス計算部261は、“@SP+”及び“@-SP”以外のアドレッシングモードに従って3入力加算器267でオペランドのアドレス計算を行なう。

【0072】ASP 262は@SP+, @-SPによりオペランドサイズが加減算された場合の他、ENETR命令またはEXITD命令でlsize値またはadjst値でSPが補正される場合にも新しい値に更新される。なお、図16では図を単純化するため、図22中でAGU内と記述された部分はアドレス生成部153に、IU内と記述された部分は整数演算部155にそれぞれ配置して接続関係を記載している。

【0073】アドレッシングモードで指定されたディスプレイメント値は命令デコード部151からDISPバス165によりアドレス計算部261へ転送されてディスプレイメントレジスタ(DISP)263に保持される。サブルーチンジャンプ命令の戻り先番地の計算、あるいは分岐すると予測した条件分岐命令の非分岐側命令アドレスの計算のため、DISP 263には命令デコード部151から転送された命令コード長(補正值)を入力することも可能である。スケールドインデックスアドレッシングに対しては整数演算部155内のレジスタからIXバス170を通じて転送された値がインデックスレジスタ(INDEX)264に入力されて保持される。INDEX 264では入力された値の1, 2, 4, 8倍を3入力加算器267へ出力することが可能である。多段間接アドレッシングにより前段までのアドレス計算結果を次の多段間接アドレッシングのベースアドレスとする場合は、3入力加算器267の出力がアドレ

ス出力レジスタ(A0)266 からベースレジスタ(BASE)265へ転送される。レジスタ相対モード、PC相対モード、レジスタベース多段間接モード、PCベース多段間接モードではベースアドレスとして指定された汎用レジスタあるいはPCの値がIXバス170からINDEX 264, 3入力加算器267, A0 266 を通じてBASE 265にロードされる。また、メモリ間接アドレッシングでは、3入力加算器267 のアドレス計算結果をA0 266からOA転送部268 を経てAAバス160へ出力し、DDバス164 から整数演算部155 のSDレジスタ291 にメモリからアドレスをフェッチし、S1バス10を経由してBASEレジスタ265 に目的アドレスを転送する。DISP 263, INDEX 264, BASE 265に保持された3つの値は3入力加算器267で加算されてA0 266へ出力される。

【0074】アドレスの計算結果はA0 266からAAバス160へ出力され、オペランドをアクセスするアドレスとして使用される。また、A0 266から出力されたオペランドアドレス自体はOA転送部268へも渡され、パイプライン中の命令の流れに同期してOA転送部268で管理される。即値も命令デコード部151からDISP 263, 3入力加算器267, A0 266を経由してOA転送部268に入力され、オペランドアドレスと同様にパイプライン中の命令の流れに同期してOA転送部268で管理される。32ビットディスプレイメント付きPC相対と絶対とを除くアドレッシングモードのジャンプ命令のジャンプ先アドレスの計算結果はA0 266からJAバス171へ出力され、パイプラインの第3ステージであるアドレス生成段階での先行ジャンプに使用される。

【0075】条件分岐命令の予測分岐先と反対側のアドレスの計算結果は分岐予測が誤っていた場合に備えて計算するものであり、条件分岐命令実行後にPC生成部154を初期化するために使用される。SP先行更新部260は”@SP+”と”@-SP”とのアドレッシングモードに対して専用の作業用SPであるASP 262でSP値の更新を行ない、更新したSP値をパイプライン中の命令の流れに同期してSP転送部270内で管理する。また、ASP 262はENETR命令またはEXITD命令の実行に伴ってlsize値またはadjst値でSPが補正される場合にも新しい値に更新される。

【0076】アドレス計算部261がSPの値を参照する場合はIXバス170を経由してASP 262の値を参照する。従って、本発明のデータ処理装置100では”@SP+”または”@-SP”モードでSP値を更新する命令、あるいはENETR命令またはEXITD命令に引き続く命令がパイプラインストールなしにSP値を用いたアドレス計算を行なうことが可能である。命令実行段階でSPに書き込みが行なわれた場合、同時にD1バス14またはD3バス15から更新する値がASP 262とSP転送部270との命令実行段階作業用SP(ESP)に書き込まれる。命令実行段階でジャンプが発生してパイプラインがクリアされた場合には、ESPの値がEP0バスを通じてASP262へ転送される。

【0077】(3.7)「オペランドアクセス部」

オペランドアクセス部156のブロック構成図を図23に示す。なお、この図23には命令フェッチ部150、アドレス生成部153、整数演算部155及びバスインターフェイス部157も共に示されている。

【0078】オペランドアクセス部156には、4KBのデータバッファメモリ290, 2エントリのオペランドプリフェッチキュー(SD)291及び(SDA)294, ストアバッファ(SBDATA)292, 整数演算部155とのデータ入出力回路(DDR)295及び(DDW)296, オペランドブレイクチェック回路(OBA)297, DMADDRレジスタ298等が備えられている。DDR 295は整数演算部155へリードデータを転送するためのレジスタである。DDW 296は整数演算部155がライトデータを出力するためのレジスタである。オペランドアドレスレジスタ299は、オペランドアクセス部156がアクセスされる場合にAAバス160から与えられるオペランドアドレスを一旦保持し、OAバス172へ出力する。

【0079】OAバス172は命令フェッチ部150にも接続しており、命令フェッチ部150のIOADDR, IOMASK, NCADDRにより、オペランドアクセスがI/O領域あるいは非キャッシュ領域に入るか否かのチェックを行なう。データバッファメモリ290はモード切り替えにより、16byte×256entryのデータキャッシュ、4KBの内蔵データRAMあるいは1Kステップの実行命令アドレス用トレースメモリの内のいずれか1つとして動作する。

【0080】データのリード動作に際しては、アドレス生成部153あるいは整数演算部155から出力されたリードすべきデータアドレスがオペランドアドレスレジスタ299へ転送されることにより、データバッファメモリ290からDDバス164を通じてデータがフェッチされてSD 291あるいはDDR295へ転送される。SD 291へデータをフェッチする場合は、ストアオペランドとのオーバーラップチェックのためにフェッチデータの8バイト境界ごとのデータアドレスがSDA291へ転送されて保持される。

【0081】データバッファメモリ290をデータキャッシュとして使用する場合は、OAバス172から転送されたデータアドレスの下位12ビットに従ってキャッシュアドレスタグとデータとが出力される。キャッシュアドレスタグはデータアドレスの上位20ビットと比較され、両者が一致すればデータキャッシュはヒットとなる。キャッシュミスした場合は、OAバス172からバスインターフェイス部157へデータアドレスが出力され、外部のメモリ(主記憶装置103)をブロック転送でアクセスしてBDバス163からデータをフェッチし、データキャッシュのエントリを更新する。ブロックリードはオペランドとして必要なデータからアドレスを昇順にラップアラウンドして行なわれ、オペランドとして必要なデータはデータキャッシュへの登録と並行してDDバス164からSD 291またはDDR 295へ転送される。

【0082】データバッファメモリ290を内蔵データRA

Mとして使用する場合は、DMADDRレジスタ298の内容とデータアドレスの上位20ビットとが比較され、両者が一致すれば下位12ビットのアドレスに従ってデータバッファから読み出されたデータが有効となる。両者が一致しない場合は外部のメモリ（主記憶装置103）がアクセスされる。

【0083】また、データバッファメモリ290には16バイトのブロックバッファ290Bが備えられており、NCADDRレジスタ300で指定される非キャッシュ領域のアクセスも含めてデータアクセスがブロックバッファ290Bのデータにヒットするか否かをチェックする。ブロックバッファ290BはNCADDRレジスタ300で指定される非キャッシュ領域に対する8バイトデータリードを16バイト一括してブロック転送により行ない、リードした16バイトのデータを保持して次のデータリードが同一の16バイト境界内部である場合にはブロックブロックバッファ290Bからデータを出力する。この機能により、ストリング操作命令あるいはビットマップ命令で非キャッシュ領域のデータを16バイト単位で高速にアクセスすることが可能となる。

【0084】データのストア動作に際しては、アドレス生成部153からAAバス160へ出力されたストアすべきデータのアドレスがオペランドアドレスレジスタ299へ転送されることにより、DDW 296から出力されたデータがDDバス164を通じて転送される。データバッファメモリ290を内蔵データRAMとして使用し、且つその領域にアクセスする場合以外の場合には、ストア動作は必ず外部のメモリ（主記憶装置103）に対して行なわれる。ストアデータアドレスはバスインターフェイス部157のSBADDR 183へ転送され、ストアデータはストアバッファ(SBDAT A)292へ転送される。DDW 396からストアバッファ(SBD ATA)292への転送時にSD 291あるいはデータキャッシュにストアデータとオーバーラップするデータが保持されている場合にはその内容が書き換えられる。SD 291のオーバーラップチェックはSDA294で、データキャッシュのオーバーラップチェックはキャッシュ内のタグでそれぞれ行なわれる。なお、データキャッシュはライトスルー方式であり、ライト動作でミスした（オーバーラップがない）場合にはキャッシュの内容は変化しない。

【0085】内蔵データRAMへのストア動作ではストアアドレスの上位20ビットがアドレスデコード機構のDMADDR 298と比較され、一致する場合は下位12ビットのアドレスに従ってストアデータが内蔵データRAMに書き込まれる。

【0086】バスインターフェイス部157がストアバッファ(SBDATA)292のデータを外部にライト中もオペランドアクセス部156は引き続きリードアクセスまたはライトアクセスを受け付ける。従って、ストアバッファ(SBD ATA)292に未処理のデータが存在している場合にも、データキャッシュがヒットしたり、あるいは内蔵データRAM

Mがアクセスされる場合はオペランドアクセス部156は後続の処理を続けることができる。

【0087】オペランドのリードあるいはライト、メモリ間接アドレッシングのためのリードに際しては全てI/O領域に入るか否かのチェックが行なわれる。I/O領域に対するメモリ間接アクセスはアドレス変換例外となる。また、I/O領域からのオペランドプリフェッチはパイプライン中の先行命令の実行が全て完了するまで抑止される。また、条件分岐命令(Bcc, ACB, SCB)に引き続く命令のオペランドプリフェッチのためのメモリアccessが内蔵データRAM領域以外へのアクセスであったり、あるいはキャッシュミスを起こした場合は、先行する条件分岐命令の実行が完了するまで外部メモリ（主記憶装置103）へのアクセスが抑止される。

【0088】ERR, INF 185はオペランドアクセスがEITを発生した場合のエラーアドレス、IOINF等を保持するブロックである。本発明のデータ処理装置100がバススヌープ動作中は、無効化すべきデータのアドレスがバスインターフェイス部157からBAバス162を経由してオペランドアクセス部156へ転送される。データキャッシュはこのアドレスがヒットする16バイトブロックのデータを全て無効化する。データバッファメモリ290を内蔵データRAMとして動作させる場合はバススヌープ動作が行なわれないので注意が必要である。

【0089】(3.8)「ROM部」

ROM部152には整数演算部155の制御を行なう種々のマイクロプログラムルーチンが格納されているマイクロROM 320、マイクロシーケンサ321、マイクロ命令デコーダ322が備えられている。ROM部152では、命令デコード部151から出力されたRコード226と副Rコード227とに従ってマイクロプログラムにより整数演算部155の動作を制御する。マイクロシーケンサ321は命令実行に関するマイクロプログラム実行のためのシーケンス処理の他に、例外、割込、トラップ(EIT)の受け付けと各EITに対応するマイクロプログラムのシーケンス処理も行なう。

【0090】ROM部152のブロック構成図を図24に示す。なお、この図24には命令デコード部151及び整数演算部155も共に示されている。Rコード226の入力ラッチはマイクロエントリ番地レジスタ(RADDR)323とパラメータレジスタ(RPARM)324とで2エントリのキューとして構成されている。副Rコード227はRコード226のパラメータの一部として扱われる。EITデコーダ(EITDEC)325はEITの種類に応じて対応するマイクロプログラムルーチンのエントリ番地を出力するデコーダである。レジスタ(PREEIT)326は命令実行途中でEITを受け付けた場合に、命令に依存したEIT前処理を行なうマイクロプログラムルーチンのエントリ番地を保持するためのレジスタである。このPREEIT 326には、任意長ビットフィールド命令、ストリング操作命令等の命令実行途中でEITを受け付ける命令を実行する際に命令依存の番地がセ

ットされる。

【0091】スタック(USTACK)327 は2エントリのマイクロプログラム用のスタックであり、マイクロプログラムのサブルーチンからの戻り先番地を保持する。戻り先番地はインクリメンタ(UEINCR)328 からセットするサブルーチンコール命令の次番地である場合とマイクロプログラムでラッチ(URDR)329 から明示的にセットする番地である場合とがある。ラッチ(UNAR)330 はマイクロROM 320 のXデコーダ入力となる10ビットを保持するラッチであり、ラッチ(μTEST)331はマイクロROM 320 のYデコーダ入力となる2ビットを保持するラッチである。

【0092】UEINCR 328はUNAR 330とμTEST 331とを連結した12ビットの値をインクリメントするインクリメンタである。URDR 329はマイクロ命令の出力ラッチである。マイクロプログラムの条件ジャンプをディレイスロットなしで行なうためマイクロプログラムの条件ジャンプ時はマイクロROM 320 から4ワードを一度に読み出し、条件判定結果に従ってμTEST 331で指定される2ビットでその内の1つを選択するようになっている。

【0093】マイクロROM 320(IROM) は155 ビット×4 KワードのROM である。マイクロ命令デコーダ322 はURDR 329から出力されるマイクロ命令とRPARM 324 から転送されたパラメータEPARM 332 とをデコードして整数演算部155 の制御信号を出力する。命令デコード部151 で2命令が同時にデコードされた場合、先行命令のデコード結果はRコード226 として出力され、後続命令のデコード結果は副Rコード227 としてRPARM 324 中に含まれてマイクロ命令デコーダ322 に入力される。副Rコード227 は先行命令の最終マイクロ命令と共にマイクロ命令デコーダ322 によりデコードされ、後述する副ALU 27及び副バレルシフタ352 を使用して実行される。

【0094】マイクロ命令はマイクロROM 320 から1クロックに1度の割合で読み出され、1つのマイクロ命令で1つのレジスタ間演算が行なわれる。従って、転送、比較、加算、減算、論理演算等の基本命令は1クロックで終了する。基本命令のRPARMO:1に副Rコード227 が含まれる場合には、1クロックにRコード226 に対する基本命令と副Rコード227 とに対するレジスタ間演算命令が実行され、命令実行速度が2命令/クロックとなる。命令実行途中でEIT が受け付けられた場合はPREEIT 326 の値がUNAR 330へ転送され、マイクロプログラムがPREEIT 326に保持されたマイクロ番地へジャンプしてEIT の前処理が行なわれ、その後にEITDEC 325から出力される各EIT の処理ルーチンにジャンプする。命令完了段階でEIT が受け付けられた場合は、PREEIT 326は使用されず、直接EITDEC 325から出力される EIT処理ルーチンへジャンプする。

【0095】(3.9)「整数演算部」

整数演算部155 は ROM部152 のマイクロROM 320 に格納されたマイクロプログラムにより制御され、各整数演算

命令を実行するために必要な演算をレジスタファイル20と演算器とを用いて実行する。整数演算部155 のブロック構成図を図25に示す。

【0096】EJUMP 355 はパイプラインの命令実行ステージ404 でジャンプを実行する際にジャンプ先アドレスを格納するレジスタである。EIT を検出したりプリリターン先アドレスを誤ったサブルーチンリターン命令を実行した場合、マイクロプログラムにより本レジスタにジャンプ先アドレスを入力して実行ステージジャンプを行う。実行ステージジャンプが行なわれた場合には、パイプラインがフラッシュされるため再度パイプラインを充填するために3~4クロックの無駄時間が発生する。

【0097】副演算回路351 には副ALU(SALU) 27と副バレルシフタ352 とが備えられており、主演算回路350 には主ALU(MALU)23,主バレルシフタ29,乗算器353,プライオリティエンコーダ354,カウンタ等が備えられている。副演算回路351 と主演算回路350 とは汎用レジスタ20G 及び作業用レジスタ20W とそれぞれ3本の32ビットバスで結合されており、2つのレジスタ間演算を同時に実行することが可能である。S1バス10, S2バス11, D1バス14 が主演算回路350 用のバスであり、S3バス12, S4バス13, D3バス15が副演算回路351 用のバスである。命令デコード部151 で同時にデコードされた2つの命令は副演算回路351と主演算回路350 とで同時に実行される。また、高機能命令を実行する場合には、マイクロプログラムにより副演算回路351 と主演算回路350 とを並列に動作させ、8バイトデータを一度に処理したり、2つのオペレーションを並列に行なうこともできる。

【0098】AA部356 とDD部357 とは整数演算部155 がオペランドアクセス部156 との間で通信を行なうためのレジスタであり、AA部356 がアドレスを、DD部357 がデータをそれぞれ扱う。AA部356 には2つのインクリメント/デクリメント機能付きアドレスレジスタと、ベースアドレスとビットオフセットとから操作対象のビットのバイトアドレスを計算するビットアドレス計算回路とが備えられている。DD部357 にはリード用に2つ、ライト用に1つの計3つの8バイトのデータレジスタが備えられており、DDバス164 によりオペランドアクセス部156 との間で8バイト単位でデータをやりとりする。

【0099】SP部358 には命令実行ステージ用の作業用スタックポインタ(ESP) と各リングに対応したスタックポインタ(SPI, SPO, SP1, SP2, SP3) とが備えられている。命令実行ステージ404 のSP値はアドレス生成部(AGU)153 から命令に同期してパイプライン中を転送されてくる。ESPにはAAバス160 への出力経路が備えられており、AA部356 を介さずにESP の値でメモリをアクセスすることができる。制御レジスタ359 にはPSW, DMC等の各種の制御レジスタと定数ROM とが備えられている。

【0100】汎用レジスタ20G にはスタックポインタであるR15 以外の14本の汎用レジスタが備えられている。

汎用レジスタ20G は、主演算回路350 と副演算回路351 とに2つずつ計4つのデータを同時に出力し、2つの演算結果を同時に書き込むため、4つの出力ポートと2つの入力ポートとを有する。更に、汎用レジスタ20G は、オペランドのアドレス計算のためにデータをアドレス生成部153 へ転送するIXバス170 に接続するもう1つの出力ポートを有する。作業用レジスタ20W は全部で16本あり、汎用レジスタ20G と同じく主演算回路350 と副演算回路351 とに接続するため4つの出力ポートと2つの入力ポートとを有する。

【0101】図26に ROM部152 のマイクロプログラムで操作する整数演算部155 及びその周辺のブロック図を示す。整数演算部155 では命令の実行あるいはEIT 処理のために、S1, S2, D1, D3バス10, 11, 14, 15を通じて各周辺ブロックにアクセスする。PC転送部242 からは命令のPC値が転送されてくる。EIT により実行中命令のPC値あるいは次命令のPC値が必要な場合はPC転送部242 からそれらの値を読が出される。RTS, EXITD命令を実行する際にはPCスタック243 からプリリターンアドレスが読み出され、メモリからフェッチされた真のリターン先アドレスと比較され、プリリターンが正しいアドレスへ行なわれたか否かがチェックされる。PC転送部242 にはデバッグサポート用レジスタIBAO:1, TGIAも備えられている。

【0102】OA転送部268 からはアドレス生成部153 で計算されたオペランドアドレスあるいは即値が読み出される。ASP 262 はオペランド生成部の作業用スタックポインタであり、命令の実行あるいは EIT処理により実行ステージ404 のスタックポインタが書き換えられた場合には図25のSP部358 のスタックポインタと同時に書き換えられる。

【0103】図26のMAU 370 は命令フェッチ部150 とオペランドアクセス部156 との両方のアドレスデコード機構を含む。MAU 370 にはメモリアクセス関連のレジスタ(IOADDR, IOMASK, NCADDR, DMADDR, OBAO:1)及びメモリアクセス関連の EIT情報保持用のレジスタ(EITADDR, IOINF)が備えられており、マイクロプログラムによりこれらのレジスタにアクセス可能である。SD 291からはプリフェッチオペランドが出力される。32ビット以下のプリフェッチオペランドはS1バス10またはS2バス11のいずれへも出力することが可能である。64ビットのプリフェッチオペランドを出力する際はS1バス10とS2バス11とを連結して使用する。SBAO:1 371とSBDO:1 372とはストアバッファ(SBDATA)292 のアドレスとデータとを保持するレジスタであり、ストアバッファ(SBDATA)292 でEIT が発生した場合にその内容をマイクロプログラムで読み出す。

【0104】(4)「本発明のデータ処理装置におけるハードウェアの動作」

本発明のデータ処理装置100 におけるハードウェアの動

作を以下に述べる。

(4.1)「パイプライン処理」

(4.1.1)「パイプラインの構成」

本発明のデータ処理装置100 における命令パイプラインの概念を図27の模式図に示す。命令のプリフェッチを行なう命令フェッチステージ (IFステージ)400, 命令のデコードを行なうデコードステージ (Dステージ)401, オペランドのアドレス生成を行なうアドレス生成ステージ (Aステージ)402, マイクロROM 320 のアクセス (特にRステージと称す) とメモリオペランドのプリフェッチ (特にOFステージと称す) とを行なうオペランドフェッチステージ (Fステージ)403, 命令の実行を行なう実行ステージ (Eステージ)404, メモリオペランドのストアを行なうストアステージ (Sステージ)405の6段のステージでパイプライン処理が行なわれる。

【0105】IFステージ400 には32バイトの命令プリフェッチキューが、Fステージ403 には2エントリのオペランドプリフェッチキュー291 が、Sステージ405 には2エントリのストアバッファ(SBDATA)292 がそれぞれ含まれる。各ステージは他のステージとは独立に動作し、理論上は6つのステージが完全に独立して動作する。Sステージ405 以外の各ステージは1回の処理を最小1クロックで行なう。Sステージ405 は1回のオペランドストア処理を、ストアバッファ(SBDATA)292 に空きがある場合は1クロックで行ない、空きがない場合は最小2クロックで行なう。従って、本発明のデータ処理装置100 の命令パイプラインは理想的には1クロックごとに次々に命令を処理する。

【0106】本発明のデータ処理装置100 の命令にはメモリーメモリ間演算、あるいはメモリ間接アドレッシング等のような1回の基本パイプライン処理では処理が行なえない場合もあるが、本発明のデータ処理装置100 の命令パイプラインではこれらの処理に対してもなるべく均衡したパイプライン処理が行なえるように構成されている。複数のメモリオペランドを有する命令に対しては、メモリオペランドの数を基にデコード段階で1つの命令を複数のパイプライン処理単位 (ステップコード) に分解してパイプライン処理する。

【0107】各パイプラインステージの入出力ステップコードには図27に示したように便宜上名前が付けられている。また、各ステップコードはオペレーションコードに関する処理を行ない、マイクロROM 320 のエントリ番地及びEステージ404 に対するパラメータとなる系列と、Eステージ404 の処理対象のオペランドとなる系列との2系列がある。また、Dステージ401 からSステージ405 の間では処理中命令のプログラムカウンタ値が受け渡され、Aステージ402 からEステージ404 の間ではスタックポインタ値が (更には、スコアボードレジスタ値も) 受け渡される。

【0108】IFステージ400 からDステージ401 に渡さ

れる情報は命令コードそのものである。Dステージ401からAステージ402に渡される情報は命令で指定された演算に関する情報(Dコード222と称す)と、オペランドのアドレス生成に関する情報(Aコード220と称す)と、処理中の命令のプログラムカウンタ値(PC値)との3つである。Aステージ402からFステージ403に渡される情報はマイクロプログラムルーチンのエントリ番地及びマイクロプログラムへのパラメータを含むRコード226、オペランドのアドレスとアクセス方法指示情報とを含むFコード225、それに処理中の命令のプログラムカウンタ値及びスタックポインタ値の4つである。Fステージ403からEステージ404に渡される情報は演算制御情報とリテラルとを含むEコード224と、オペランド及びオペランドアドレスを含むSコード228と、それに処理中の命令のプログラムカウンタ値(PC値)及びスタックポインタ値の4つである。Sコード228はアドレスとデータとで構成される。Eステージ404からSステージ405に渡される情報は、ストアすべき演算結果であるWコード229及びその演算結果を出力した命令のプログラムカウンタ値(PC値)の2つである。Wコード229はアドレスとデータとストア関連情報とで構成される。

【0109】Eステージ404が本来の命令実行ステージである。他のIF、D、A、Fステージは命令実行のための前処理を行なうステージであり、命令コードあるいはオペランドの読み出しは行なうがメモリあるいはレジスタの内容の変更は一切行なわない。このため、これらのIF、D、A、Fステージでの処理はEステージ404の処理結果に依存してキャンセルされる場合がある。

【0110】(4.1.2)「命令フェッチステージ」

命令フェッチステージ400(IFステージ)では命令フェッチ部150が動作する。IFステージ400では、内蔵命令キャッシュ200または外部のメモリ(主記憶装置103)から命令をフェッチして命令キュー201に取込み、Dステージ401に対して2~8バイト単位で命令コードを出力する。命令キュー201への入力、命令キャッシュ200がヒットした場合は整置された16バイト内の8バイト単位で、ミスした場合は整置された4バイト単位でそれぞれ行なわれる。命令キュー201は条件分岐命令に引き続く命令と分岐先命令との両方をフェッチするため、図18に示されているように、AとB(201Aと201B)との2つが備えられている。

【0111】単一転送により外部のメモリから命令をフェッチする場合には整置された4バイトにつき最小2クロックを要する。バーストモードによるブロック転送では16バイトにつき最小5クロックを要する。命令を外部からフェッチする際には、フェッチ先命令のアドレスがNCADDRレジスタ300で指定された非キャッシュ領域であるか否か、IOADDR、IOMASKで指定されたI/O領域であるか否かのチェックも行なわれる。なお、I/O領域からの命令フェッチは禁止されており、EITとなる。

【0112】内蔵命令キャッシュ200がヒットした場合は、整置された16バイト内の任意の8バイトの命令が1クロックでフェッチされる。命令キュー201の出力単位は2バイトごとに可変であり、1クロックの間に最大8バイトまで出力される。内蔵命令キャッシュ200の制御、プリフェッチ先命令アドレスの管理及び命令キュー201の制御もIFステージ400で行なう。IFステージ400で行なわれる処理を以下にまとめて示す。

【0113】・命令コードのプリフェッチ及びDステージ401への出力

- ・2本の命令キュー(A, B)201の管理
- ・命令プリフェッチアドレスのインクリメント
- ・命令キャッシュ200の管理
- ・命令のアドレスが非キャッシュ領域に入るか否かのチェック
- ・命令のアドレスがI/O領域に入るか否かのチェック

【0114】(4.1.3)「命令デコードステージ」

命令デコードステージ401(Dステージ)では命令デコード部151とPC生成部154とが動作する。Dステージ401ではIFステージ400から入力された命令のオペレーションコードの前段デコードとアドレッシングモードのデコードとを行なう。Dステージ401での命令デコード処理では、命令デコード部151のメインデコーダ211、第1サブデコーダ212、第2サブデコーダ213、アドレッシングモードデコーダ214、拡張データ処理部215が動作する。デコードは1クロックに1度の割合で行なわれ、1回のデコード処理で0~8バイトの命令コードが消費される(サブルーチンリターン命令の復帰先アドレスを含むステップコードの出力処理等では命令コードは消費されない)。2つのメモリオペランドを有する命令または多段間接モードを使用する命令等のような1回のデコード処理ではデコードできない命令はこのDステージ401で複数のステップコードに分解される。このため、命令全体のデコードが完了しなくても各クロックのデコード処理でDコード222とAコード220と命令のPC値とが出力される。

【0115】Dステージ401では命令デコード処理に伴って消費した命令長を基にPC生成部154で各命令のPC値を計算する。条件分岐命令、絶対アドレスまたはPC相対アドレスへのジャンプ命令(サブルーチンジャンプ命令を含む)及びサブルーチンリターン命令に対してはPC生成部154で先行ジャンプ処理(Dステージジャンプ)を行なう。無条件ジャンプ命令をデコードして先行ジャンプを行なった場合は、IFステージ400に対して、二つの命令キュー200A、200Bの内の現在使用中の命令キューの内容をキャンセルしてジャンプ先から命令をフェッチしてキューに入力し、そのコードを命令デコード部151へ出力することを指示する。条件分岐命令をデコードした場合は、分岐予測には関係なくIFステージ400に対して二つの命令キュー200A、200Bの内の現在使用中のキュー

はそのままにしておき、もう一方の命令キューに分岐先から命令をフェッチして入力することを要求する。分岐すると予測する条件分岐命令では更に新規のキューから命令コードを命令デコード部151へ出力することを指示する。

【0116】1回のデコード処理でAステージ402に対してアドレス計算情報であるAコード220、オペレーションコードの中間デコード結果であるDコード222及び命令のPC値を出力する。Dステージ401の処理を以下にまとめて示す。

【0117】・命令のオペレーションコードの前段デコード

- ・アドレッシングモードのデコード
- ・命令のステップコードへの分解
- ・PC値の計算
- ・条件分岐命令の分岐先アドレス計算
- ・条件分岐命令の分岐予測及び先行分岐処理
- ・絶対またはPC相対アドレスへの先行ジャンプ処理
- ・サブルーチンリターン命令の先行リターン処理

【0118】(4.1.4)「アドレス生成ステージ」

アドレス生成ステージ402(Aステージ)では命令デコード部151とアドレス生成部153とが動作する。Aステージ402では、Dステージ401から転送されたDコード222をデコードしてRコード226を出力し、またAコード220に従ってオペランドアドレスの生成を行なってFコード225を出力する。Dステージ401から転送されてきたPC値はそのままFステージ403へ転送される。また、命令実行ステージ404とは独立に、処理中の命令のスタックポインタ値を計算してFステージ403へ転送する。

【0119】Aステージ402での命令デコード処理では、命令デコード部151のAステージデコーダ216とサブコード転送部217とが動作する。Dステージ401から出力されたオペレーションコードの中間デコード結果は更に詳細にデコードされてマイクロROM 320のエントリアドレスとマイクロプログラムへのパラメータとがRコード226として出力される。

【0120】アドレス生成部153はAコード220に従ってオペランドアドレスを生成する。レジスタ間接モード及びレジスタ相対モードに対してはIXバス170を通じて汎用レジスタ20Gの値を参照する。ディスプレイメント値、即値、絶対アドレス値はAコード220の一部として命令デコード部151からDISPバス165を通じて転送されてくる。メモリ間接モードに対しては、生成した間接アドレスに対応して発生するステップコードがFステージ403、Eステージ404を介してメモリから目的とするアドレス値をフェッチするまで処理待ち状態となる。PC相対モードではDステージ401から転送された処理中命令のPC値(APC)が用いられる。スタックポップモード及びプッシュモードに対しては、専用の作業用スタックポインタでSP値が計算される。計算されたSP値はFコード22

5と共にFステージ403へ出力される。

【0121】レジスタ間接ジャンプまたはメモリ間接ジャンプ等の絶対ジャンプとPC相対ジャンプ以外のジャンプ命令とに対しては、アドレス生成部153で飛び先アドレスが計算されて先行ジャンプ処理(Aステージジャンプ)が行なわれる。サブルーチンジャンプ命令の戻り先アドレスの計算はDISPバス165から転送されてきた命令長とAPC値とを加算することにより行なわれる。条件分岐命令に対しては、分岐予測が誤った場合にPC生成部154を初期化するために分岐予測側アドレスと反対側の命令アドレスを計算する。

【0122】アドレス生成部153にはアドレス計算に使用するレジスタあるいはメモリがコンフリクトしてデータハザードが発生した場合にそれを検出してパイプラインをインターロックするためのスコアボードレジスタ480が備えられている。Aステージ402ではスコアボードレジスタ480へのデスティネーションオペランドの登録(書き込み予約)及びアドレス生成のためにレジスタあるいはメモリからデータを読み出す際のデータハザードのチェックも行なわれる。スコアボードレジスタ480には16個の汎用レジスタに対応してそれぞれ1個のフラグが用意されている。本発明のデータ処理装置100では多段間接モードを使用しているため、Dステージ401で複数のステップコードに分解された命令はAステージ402でオペランドのアドレスが計算されて1つのステップコードに縮退する。

【0123】レジスタリストを用いて転送オペランドとなるレジスタ番号を示す命令(LDM, STM, ENTER, EXITD)はAステージ402で複数のステップコードに分解される。これらの命令はAステージ402で1つまたは2つのレジスタのデータを転送するステップコードに分解されてF, E, Sステージでパイプライン処理される。分解された各ステップコードで転送すべきオペランドのレジスタ番号はAステージデコーダ216でレジスタリストをエンコードすることにより得られる。また、各ステップコードでアクセスするメモリのアドレスはAステージデコーダ216から出力される制御信号(AAコード)に従ってアドレス生成部153が生成する。Aステージ402の処理を以下にまとめて示す。

【0124】・命令のオペレーションコードの後段デコード

- ・オペランドアドレスの生成
- ・スタックポインタ値の計算
- ・条件分岐命令の分岐予測と反対側の命令アドレスの計算
- ・サブルーチンジャンプ命令の戻り先アドレスの計算
- ・レジスタ間接、レジスタ相対等のアドレッシングモードのジャンプ命令に対する先行ジャンプ処理
- ・スコアボードレジスタ480へのデスティネーションの書き込み予約

・アドレス生成のためのデータハザードチェックとパイプラインインターロック

・レジスタリストを有する命令のステップコードへの分解

【0125】(4.1.5)「オペランドフェッチステージ」オペランドフェッチステージ403(Fステージ)ではROM部152とオペランドアクセス部156とが動作する。Rコード226に対するROM部152のマイクロROM320アクセス動作とオペランドアクセス部156の動作とは独立した制御の下で行なわれる。これらの2つの動作を別々に扱う場合、ROM部152でのマイクロROM320へのアクセス処理を特にRステージの処理と称し、オペランドアクセス部156の処理を特にOFステージの処理と称す。RステージはRコード226を入力とし、Eコード224を出力する。OFステージはFコード225を入力とし、Sコード228を出力する。Fコード225はキューイングされないが、Rコード226は2つまでキューイングされる。Eコード224はキューイングされないが、Sコード228は2つまでキューイングされる。

【0126】Rステージの処理であるマイクロROM320に対するアクセス処理はRコード226に対して次のEステージ404での実行に際して使用される実行制御コードであるEコード224を生成するためのマイクロROM320アクセスとマイクロ命令デコード処理とである。1つのRコード226に対する処理が2つ以上のマイクロプログラムステップに分解される場合、ROM部152がEステージ404で使用され、次のRコード226がマイクロROM320に対するアクセス待ちになる場合がある。Rコード226に対するマイクロROM320アクセスが行なわれるのはEステージ404でのマイクロROM320アクセスが行なわれない場合である。本発明のデータ処理装置100では、多くの整数演算命令が1マイクロプログラムステップで完了するため、実際にはRコード226によるマイクロROM320に対するアクセスが次々と行なわれることが多い。

【0127】オペランドフェッチステージ403では、Fコード225のデータアドレスで内蔵データキャッシュ、内蔵データRAMまたは外部のメモリをアクセスしてオペランドをフェッチし、そのオペランドとFコード225のデータアドレスとを組み合わせ、Sコード228を生成して出力する。

【0128】1つのFコード225では8バイト境界をクロスしてもよいが、8バイト以下のオペランドフェッチを指定する。Fコード225にはオペランドのアクセスを行なうか否かの指定も含まれており、Aステージ402で計算されたオペランドアドレス自体あるいは即値がEステージ404へ転送される場合にはオペランドのフェッチは行なわれず、Fコード225の内容がSコード228として転送される。Sコード228は2つまでキューイングされるため、オペランドは2つまでプリフェッチできる。

単一転送により外部のメモリからオペランドをフェッチする際には整置された4バイトにつき最小2クロックを要する。バーストモードによるブロック転送では整置された16バイトにつき最小5クロックを要する。

【0129】オペランドを外部のメモリからフェッチする際にはフェッチ先のアドレスがNCADDRレジスタ300で指定された非キャッシュ領域であるか、IOADDR、IOMASKで指定されたI/O領域内であるかのチェックも行なわれる。I/O領域からはオペランドのプリフェッチは行なわない。この場合、先行命令の実行が全て完了し、I/O領域をアクセスする命令が実行されることが確実になった後にオペランドのフェッチが行なわれる。また、条件分岐命令(Bcc, ACB, SCB)または条件トラップ命令(TRAP)に引き続く命令でデータキャッシュがミスした場合、または内蔵データRAM領域以外からのデータフェッチが行なわれる場合には、先行する条件分岐命令、条件トラップが実行されるまでオペランドのフェッチは行なわれない。これは、論理的には実行され得ない命令のオペランドプリフェッチが外部に対して行なわれることを抑止するためである。

【0130】内蔵データキャッシュがヒットした場合は整置された8バイトにつき1クロックでオペランドがフェッチされる。内蔵データキャッシュ及び内蔵データRAMの制御、プリフェッチ先オペランドアドレスの管理及びSコードキューの制御もOFステージで行なわれる。以下にFステージ403の処理をまとめて示す。

【0131】・マイクロROM320のアクセス

- ・オペランドのプリフェッチ
- ・オペランドプリフェッチキュー291の管理
- ・データバッファ(キャッシュまたはRAM)の管理
- ・オペランドのアドレスが非キャッシュ領域に入るか否かのチェック
- ・オペランドのアドレスがI/O領域に入るか否かのチェック
- ・I/O領域からのプリフェッチ抑止
- ・先行する条件分岐命令、条件トラップ命令の実行完了までの後続命令の外部メモリアクセス抑止

【0132】(4.1.6)「実行ステージ」

実行ステージ404(Eステージ)ではEコード224及びSコード228を入力として整数演算部155が動作する。このEステージ404が命令を実行するステージであり、Fステージ403以前のステージで行なわれた処理は全てEステージ404のための前処理である。Eステージ404でジャンプが実行されたり、EIT処理が起動されたりした場合は、IFステージ400～Fステージ403までの処理は全て無効化される。Eステージ404はマイクロプログラムにより制御され、Rコード226で示されたマイクロプログラムルーチンのエントリ番地からの一連のマイクロ命令を実行することにより命令を実行する。

【0133】マイクロROM320の読み出しとマイクロロ

10

20

30

40

50

令の実行とはパイプライン化されて行なわれる。マイクロプログラムの条件分岐はマイクロROM 320 のYデコーダ（マイクロROM 320 から同時に読み出した4つのマイクロ命令から目的のマイクロ命令をセレクトするためのデコーダ）で行なわれるため、マイクロプログラムの条件分岐が発生した場合にも1クロックの空きもなく次のマイクロ命令が実行される。Eステージ404では、Aステージ402で行なったスコアボードレジスタ480への書き込み予約の解除、Dステージ401での分岐予測が間違っていた場合の分岐予測テーブル450の書き換え、プリリターンアドレスが正しいか否かのチェックも行なう。

【0134】各種の割り込みは命令の切れ目でEステージ404で直接受け付けられ、マイクロプログラムにより必要な処理が実行される。その他の各種EITの処理もEステージ404でマイクロプログラムにより行なわれる。演算の結果をメモリにストアする必要がある場合、Eステージ404はSステージ405へWコード229とストア処理を行なう命令のプログラムカウンタ値とを出力する。メモリへのオペランドストアは全てプログラムで論理的に指定された順序で行なわれる。

【0135】メモリへのストア処理を行なう場合、データキャッシュの書き換えの必要性を判断するため、Eステージ404でデータキャッシュのタグをチェックする。このため、オペランドストア動作では実際のストア処理の前にEステージ404が1クロックサイクルの間データキャッシュを占有する。単純な命令を実行する場合はFステージ403でオペランドをリードし、Eステージ404で演算を実行し、Sステージ405でオペランドをストアするが、ストリング操作命令あるいはビットマップ操作命令を実行する場合はEステージ404でバーストモードによるブロック転送を効率的に使用してメモリアクセスを行なう。従って、これらの命令では1ブロック=16バイトのデータをひとかたまりとして処理が行なわれる。以下にEステージ404での処理をまとめて示す。

【0136】・オペレーションコードで指定された命令オペレーションのマイクロプログラムによる実行

- ・汎用レジスタ値、制御レジスタ値の更新
- ・PSW中のフラグの更新
- ・リング毎に指定されたスタックポインタの更新
- ・EIT処理の受付と実行、Eステージジャンプ
- ・スコアボードレジスタ480の書き込み予約解除
- ・分岐予測テーブル450の更新
- ・プリリターンアドレスのチェック
- ・誤った先行ジャンプを訂正するためのEステージジャンプ
- ・制御レジスタの更新、インターロック命令の実行によるEステージジャンプ

【0137】(4.1.7) 「ストアステージ」

ストアステージ405(Sステージ)ではオペランドアクセス部156が動作する。データバッファメモリ290を内蔵

データRAMとして使用する場合、Sステージ405はWコード229のアドレスに従って、そのアドレスの上位20ビットとDMADDRレジスタ298の内容とが一致すればWコード229のデータを内蔵データRAMにストアし、一致しなければWコード229のデータとプログラムカウンタ値とをストアバッファ(SBDATA)292に入力して外部のメモリにWコード229のデータをストアする。データバッファメモリ290をキャッシュとして使用する場合、Sステージ405はWコード229のアドレスに従って、Wコード229のデータを内蔵データキャッシュにストアすると同時にWコード229をストアバッファ(SBDATA)292に入力し、外部のメモリにWコード229のデータをストアする。

【0138】ストアバッファ(SBDATA)292は2エン트리あり、8バイトのデータ2つまでのストア要求をキューイングできる。単一転送では、ストアバッファ(SBDATA)292から外部のメモリへのオペランドのストアが整置された4バイトにつき最小2クロックに1回行なわれる。バーストモードのブロック転送では、ストアバッファ(SBDATA)292から外部のメモリへのオペランドのストアが整置された16バイトにつき最小5クロックで行なわれる。

【0139】Sステージ405の動作はオペランドアクセス部156で行なわれ、内蔵データキャッシュがミスした場合も入れ替え処理は行なわれない。オペランドのストア処理でEITが検出された場合は、ストアバッファ(SBDATA)292にWコード229を保持したままでEステージ404にEITが通知される。ストアバッファ(SBDATA)292でEITが発生した場合、ヒットしたデータキャッシュのエントリは既に書き変わっていて次の命令で使用される可能性があるため、注意が必要である。また、そのヒットしたデータキャッシュのエントリはストアバッファ(SBDATA)292のEITによりハードウェアで自動的に無効化されることはないため、この点も注意が必要である。以下にSステージ405で行なわれる処理を示す。

【0140】・データバッファメモリ290と主記憶装置103へのストア

- ・ストアバッファ(SBDATA)292の管理

【0141】(4.2) 「スーパースケーラ処理」

(4.2.1) 「スーパースケーラ処理の概要」

本発明のデータ処理装置100は高頻度で実行する短縮フォーマットの命令を先行する命令と並列してデコードし、並列して実行する2命令のスーパースケーラ処理を行なう。パイプライン処理とスーパースケーラ処理とが共に最高効率で動作する場合、本発明のデータ処理装置100は1クロックに2命令を処理し、40MHz動作時に最大80MIPSの処理能力がある。

【0142】2命令を並列してデコードする場合、図20に示した命令デコード部151のメインデコーダ211が先行命令をデコードし、第1サブデコーダ212または第2

10

20

30

40

50

サブデコーダ213 が後続命令をデコードする。また、2 命令を並列して実行する際には、図25に示した整数演算部155 では主演算回路350 が先行命令を実行し、副演算回路351 が後続命令を実行する。また、2 命令を並列して実行する際には、副演算回路351 はハードワイヤード制御により後続命令を実行する。先行命令が2 マイクロ命令以上のマイクロプログラムで実行される場合は、後続命令は最終マイクロ命令が実行される際に並列して実行される。

【0143】図28の模式図にパイプライン中で2 命令のスーパースケラ処理を行なう場合のタイミングの一例を示す。この例では、命令フェッチ (IF)、命令デコード (D)、アドレス生成 (A)、オペランドフェッチ (F)、命令実行 (E)、オペランドストア (S) の各パイプラインステージで「MOV R3,R1」と「SHL #2,R3」、「ADD R3,R2」と「ADD R5,RO」、「MOV @SP+,R4」と「ADD R3,RO」の3組の命令が並列処理されている。これらの命令の組合せでは、後続命令がレジスタオペランドのみの命令であって先行命令とオペランドとの競合がないため、2 命令の並列デコード、並列実行が行なわれる。

【0144】本発明のデータ処理装置100 の命令フェッチ部150 には16バイトの命令プリフェッチキューがあり、命令キャッシュ200 と命令キュー201 とは8バイトのバスで結合されているため、命令を一度に8バイトフェッチする。図28の例では、先行する4 命令を同時にフェッチし、次に後続の3 命令を同時にフェッチした状況が示されている。なお、図28において“—”は各命令がパイプライン中でストールして処理待ち状態であることを示している。また、ハッチングはデータメモリ(データキャッシュ)に対するアクセス処理を示す。

【0145】(4.2.2) 「並列デコード処理」

本発明のデータ処理装置100 の命令デコード部151 は2 命令を並列してデコードするための構成として、メインデコーダ211 と2つのサブデコーダとを備えている。2 命令をデコードする場合、メインデコーダ211 は16ビットまたは32ビットの先行命令をデコードし(先行命令が複数のステップコードに分解される場合の最後のデコード処理も含む)、第1サブデコーダ212 はメインデコーダ211 がデコードする先行命令が16ビットであると仮定して「先行命令の番地+2番地」の命令をデコードする。また、第2サブデコーダ213 は先行命令が32ビットであると仮定して「先行命令の番地+4」の命令をデコードする。

【0146】2つのサブデコーダのデコード結果に関しては、メインデコーダ211 のデコード結果により先行命令の命令長が判明した時点で有効無効の判断がされる。図29に並列デコードが可能な命令コードの組合せ条件を示す。フェッチされた命令コードは命令キュー201 によりシフトして命令デコーダに供給されるため、図29中の先頭番地は任意の偶数番地でよい。但し、命令フェッチ

の効率の面からは、ジャンプ先命令は8バイト境界であることが望ましい。

【0147】(4.2.3) 「並列実行処理」

命令デコード部151 で並列してデコードされた2つの命令は整数演算部155 で並列実行される。先行命令が整数演算命令である場合、後続命令は先行命令の最終マイクロステップを実行する際に並列して実行される。整数演算部155 は2 命令を並列して実行するために、主演算回路350 と副演算回路351 とがそれぞれ3本のバスでレジスタファイル20に結合されている。図30に ROM部152 と整数演算部155 との並列実行機構に関する部分のブロック図を示す。

【0148】主演算回路350 は汎用レジスタ20G 及び作業用レジスタ20W で構成されるレジスタファイル20、AA部356 及びDD部357 とS1バス10、S2バス11、D1バス14で結合されており、レジスタオペランド、即値、メモリオペランド間で先行命令の演算を行なう。主演算器の内の乗算器353 のみはD3バス15にも結合されている。副演算回路351 は汎用レジスタ20G 及びSP部358 にS3バス12、S4バス13、D3バス15で結合されており、主演算回路350 と並列してレジスタオペランド間で後続命令の演算を行なう。

【0149】先行命令を実行するためのマイクロ命令がデコードされる場合、後続命令のサブRコード227 がサブコードデコーダ420 でデコードされる。副演算回路351 はサブコードデコーダ420 のデコード結果に従って、先行命令の演算に際して使用するバスとは独立のバスでレジスタオペランドをアクセスして演算を行なう。従って、後続命令の演算結果が先行命令の実行に影響を与えたり、先行命令の演算結果が後続命令の実行に影響を与えない限り、2つの演算は完全に並列実行可能である。なお、先行命令と後続命令とのオペランド間の干渉は命令デコード部151 がチェック済みであり、命令デコード部151 で並列デコードされた2の命令は常に並列実行可能である。

【0150】(4.2.4) 「並列処理が可能な命令の組み合わせ」

スーパースケラ処理を行なう場合、命令デコード部151 ではメインデコーダ211 が32ビット以下の先行命令をデコードし、第1サブデコーダ212 または第2サブデコーダ213 が16ビットの後続命令をデコードする。オペランドアドレス生成部153 は先行命令のメモリオペランドのアドレスを生成し、後続命令のオペランドアドレスは生成しない。また、オペランドアクセス部156 は先行命令のオペランドをフェッチし、後続命令のオペランドはフェッチしない。整数演算部155では、主演算回路350 がマイクロプログラム制御で先行命令を実行し、副演算回路351 がハードワイヤード制御で後続命令を実行する。

【0151】スーパースケラ処理は上述のハードウェ

ア動作の範囲内で行なわれるため、2命令の組合せ全てにおいて可能ではない。スーパースケーラ処理が可能な命令の一覧を以下に示す。先行命令はほとんど全ての命令を含むが、後続命令はメモリオペランドを含まない短縮形フォーマットの命令に限られる。また、先行命令と後続命令との順序が逆の場合には本発明のデータ処理装置100は並列処理を行なわないので注意が必要である。コンパイラは下記の条件を満たす2命令が連続するオブジェクトコードを出力し、スーパースケーラ処理の実行頻度を向上させることが望ましい。

【0152】1) 先行命令

以下の命令で多段階モード、32ビットのアドレッシング拡張部を含まない命令

転送命令	: LDM, STMを除く全命令
比較命令	: CHKを除く全命令
算術演算命令	: MULX, DIVXを除く全命令
論理演算命令	: MTSTを除く命令
シフト命令	: 全命令
ビット操作命令	: 全命令
固定長ビットフィールド操作命令	: 全命令
任意長ビットフィールド操作命令	: なし
10進命令	: UNPKを除く全命令
キュー命令	: QINS, QDEL
制御空間・物理空間操作命令	: LDC, LDPSBを除く全命令
OS関連命令	: STCTX

【0153】2) 後続命令

以下の命令でアドレッシングモードがRnでかつ先行命令とオペランドとが干渉しない命令

転送命令	: MOV:L, MOV:S, MOV:Q, MOV:Z
比較命令	: CMP:L, CMP:Q, CMP:Z
算術演算命令	: ADD:L, ADD:Q, SUB:L, SUB:Q, NEG
シフト命令	: SHA:Q, SHL:Q
論理演算命令	: AND:R, OR:R, XOR:R, NOT

【0154】(4.3) 「複数オペランドの並列処理」

(4.3.1) 「並列化整数演算部」

本発明のデータ処理装置100の整数演算部155には演算器、レジスタポート、バスがそれぞれ2組ずつ備えられており、2並列のスーパースケーラ処理を行なう他、LDM, STM, ENTER, EXITD命令あるいはコンテキストスイッチ命令のように、レジスタファイル20とメモリとの間で複数のオペランドを転送する命令を実行する場合に整数演算部155で2つの32ビットデータを並列して処理する。整数演算部155ではまた、ストリング命令あるいは

任意長ビットフィールド命令を実行する場合、16バイト単位でメモリをアクセスし、内部処理は8バイト単位で行なう。たとえば、バイトストリングの処理では16バイト単位でメモリをアクセスし、演算器ではキャラクタを8個単位で並列して処理する。

【0155】LDM, STM, ENTER, EXITD命令ではレジスタリストを命令デコード部151でエンコードし、これらの命令を1つまたは2つの4バイトオペランドを処理する内部コード(ステップコード)複数個に分解する。Fステージ403以降のパイプラインステージはこのステップコードを処理する。コンテキストスイッチ命令、ストリング命令あるいは任意長ビットフィールド命令では1つのマイクロ命令で主演算回路350と副演算回路351とを並列して制御して複数オペランドの並列処理を実現する。

【0156】複数オペランドを並列して処理する場合、本発明のデータ処理装置100の整数演算部155では主演算回路350と副演算回路351とをVLIW(Very-Long-Instruction-Word)技法を用いたマイクロプログラムで制御する。VLIW技法を用いた本発明のデータ処理装置100の各マイクロ命令語は1語が155ビットあり、その内の115ビットが主演算回路350を制御するためのフィールドであり、残りの40ビットが副演算回路351を制御するためのフィールドである。

【0157】図31のブロック図にVLIW技法を用いたマイクロ命令により制御される整数演算部155の構成例を示す。マイクロROM 320から出力されたマイクロ命令が主演算回路用マイクロ命令デコーダ430と副演算回路用マイクロ命令デコーダ431とでそれぞれデコードされる。この際、図31に示されているマルチプレクサ432はROM部152から出力されるマイクロ命令の副演算回路351用の制御フィールドを選択し、サブRコード227は選択しない。副演算回路351はVLIW技法を用いたマイクロ命令を実行する場合、サブRコード227に従ってスーパースケーラ処理を行なうのではなく、マイクロ命令の副演算回路351制御フィールドで指定された演算を行なう。

【0158】主演算回路350はレジスタファイル20からS1バス10とS2バス11とを通じてオペランドをフェッチし、主ALU 23あるいは主バレルシフタ29等の演算器で演算を行ない、D1バス14を通じて演算結果をレジスタファイル20に書き戻す。また、主演算回路350はメモリオペランドを扱う場合はS1バス10またはD1バス14を用いてDD部357との間でオペランドをやり取りする。副演算回路351はレジスタファイル20からS3バス12とS4バス13とを通じてオペランドをフェッチし、副ALU27あるいは副バレルシフタ352で演算を行ない、D3バス15を通じて演算結果をレジスタファイル20に書き戻す。また、副演算回路351はメモリオペランドを扱う場合はS2バス11またはD3バス15を用いてDD部357との間でオペランドをやり取りする。

【0159】(4.3.2)「複数オペランドの転送命令」
LDM, STM, ENTER, EXITD命令を処理する場合、本発明のデータ処理装置100ではこれらの命令をAステージ402で複数のステップコードに分解する。F, Eステージでは各ステップコードが1つまたは2つのオペランドを転送する命令として処理される。2つの32ビットオペランドをメモリからレジスタファイル20にロードする場合、Fステージ403では1つの64ビットデータをメモリからフェッチし、Eステージ404ではDD部357で2つの32ビットオペランドに分解してそれらを2つの32ビットバスで2つの汎用レジスタ20Gへ転送する。また、2つの32ビットオペランドをレジスタファイル20からメモリにストアする場合は、Eステージ404で2つの汎用レジスタから2つの32ビットオペランドをフェッチしてDD部357で連結して1つの64ビットデータとする。Sステージ405ではこの1つの64ビットデータをメモリにストアする。

【0160】2つの32ビットオペランドを並列してメモリからレジスタにロードする場合の整数演算部155の動作手順を以下に示す。

- 1) 1つの64ビットデータをメモリからフェッチし、DD部357に入力する。
- 2) DD部357で1つの64ビットデータを2つの32ビットオペランドに分解する。
- 3) 二つの32ビットオペランドの内の1つをS1バス10, 主ALU23, D1バス14を経由してDD部357からレジスタファイル20へ転送し、残りの1つをS2バス11, 副ALU27, D3バス15を経由してDD部357からレジスタファイル20へ転送する。

【0161】また、2つの32ビットオペランドを並列してレジスタからメモリにストアする場合の整数演算部155の動作手順を以下に示す。

- 1) 2つの32ビットオペランドの内の1つをS1バス10, 主ALU23, D1バス14を経由してレジスタファイル20からDD部357へ転送し、残りの1つをS2バス11, 副ALU27, D3バス15を経由してレジスタファイル20からDD部357へ転送する。
- 2) DD部357で2つの32ビットオペランドを1つの64ビットデータに連結する。
- 3) 1つの64ビットデータをDD部357からメモリにストアする。

【0162】図32の模式図にレジスタリストを有する命令の例として、LDM命令のビットパターンを示す。LDM命令は汎用アドレッシングモードで指定されたメモリ領域から複数の32ビットオペランドをレジスタリストで指定された汎用レジスタにロードする命令である。16ビットのレジスタリストの各ビットはレジスタ0(R0)からレジスタ15(R15)の各汎用レジスタ20Gに対応しており、レジスタリスト中のビット"1"の位置に対応する汎用レジスタに32ビットのオペランドがロードされる。本発

明のデータ処理装置100ではこのLDM命令で連続した2つの番号の汎用レジスタに同一の8バイト境界にある2つの32ビットオペランドをロードする場合、2つのオペランドを同時へ転送する。連続しない番号の汎用レジスタにオペランドをロードする場合はオペランドを1つずつロードする。

【0163】たとえば「LDM #SP+, (R4-R11)」命令を実行する場合、SP=H■(A+4)でAがダブルワード境界のアドレスであれば、転送対象のオペランドは図33のハッチング部分となる。この命令では4バイトのオペランド8個をR4~R11の8個の汎用レジスタにロードする命令であり、一度に1オペランドずつロードする従来の方法では図33(a)に(1)~(8)までアクセス回数を示すように8回のメモリアクセスが必要である。しかし、本発明のデータ処理装置100では、最初に4バイトのオペランド1つをR4へ転送し、次の6オペランドは一度に2オペランドづつ3回でR5~R10へ転送し、最後に1オペランドをR11へ転送するため、図33(b)に(1)~(5)までアクセス回数を示すように5回のメモリアクセスで8個のオペランドをロードする。

【0164】レジスタリストから2つ連続した"1"の位置をエンコードするための構成として、本発明のデータ処理装置100の命令デコード部151では図34に示すPLA回路440を備えている。このPLA回路440には命令で指定されたレジスタリスト441からマスク回路442により検索不要ビットをマスクした16ビットのビット列とオペランドアドレスの下位3ビットとが入力され、2オペランドの並列転送が可能であるか否かの判定が行なわれる。2オペランドを同時に並列転送可能な条件を図35にまとめる。

【0165】ここで、連続する番号のレジスタとワード境界以外のアドレスとの間で転送を行なう場合、レジスタ間のデータ転送では2つのデータを同時に取り扱うことが可能であるが、キャッシュミスが発生した場合にメモリアクセスが8バイトにつき3回になって効率が悪いことに注意すべきである。図35の条件はSTM, ENTER, EXITD命令でも同じである。本発明のデータ処理装置100の2オペランド同時転送機能を効率よく動作させるためには図35に示す2オペランド同時転送条件を考慮してLDM, STM, ENTER, EXITD命令を使用すべきである。

【0166】(4.3.3)「コンテキストスイッチ命令」
図36の模式図に本発明のデータ処理装置100のコンテキストスイッチの際に使用されるレジスタ群であるコンテキストブロックを示す。LDCTX命令を実行する際には本発明のデータ処理装置100はこの図36に示されているような隣り合う2つのレジスタを組にして8バイト単位でデータをロードする。この際、メモリアクセスはキャッシュ領域、非キャッシュ領域にかかわらずつねに16バイト境界単位のプロック転送により行なわれる。

【0167】STCTX命令を実行する場合はCTXBBレジス

タ443 が16バイト境界のアドレスを示すかそれ以外のアドレスを示すかにより動作が異なる。コンテキストブロックが16バイト境界に配置された場合、コンテキストブロック内のデータが4ワードずつブロック転送でストアされる。具体的には図36に示されているレジスタCSW, SP0, SP1, SP2の内容がまずブロック転送され、次にレジスタSP3, R0 の内容が単一転送され、更にレジスタR1, R2, R3及びR4の内容、レジスタR5, R6, R7及びR8の内容、R9, R10, R11及びR12 の内容の3組のデータがブロック転送され、最後にレジスタR13, R14の内容が単一転送される。コンテキストブロックが16バイト境界に配置されない場合は、図36中の隣り合う2つのレジスタが8バイト単位で単一転送によりストアされる。ブロック転送を使用するか8バイト単位の単一転送を使用するかはハードウェアが判断する。

【0168】CXTBFMのRG=0でコンテキストスイッチ命令でRO:14 を転送しない場合、LDCTX命令ではブロック転送がレジスタCSW, SP0, SP1, SP2, SP3 の内容について最小2回行なわれる。STCTX命令では CTXBBレジスタ443 が16バイト境界のアドレスを示すならばレジスタCSW, SP0, SP1, SP2の内容がブロック転送され、次にレジスタSP3 の内容が単一転送される。STCTX命令で CTXBBレジスタ443 が16バイト境界のアドレス以外ならばレジスタCSWとSP0, SP1とSP2 がそれぞれペアになり、8バイト単位でデータライトが2回実行され、次にレジスタSP3 の内容が4バイトデータとしてライトされる。

【0169】(4.3.4) 「ストリング命令」

本発明のデータ処理装置100 の整数演算部155 では、8, 16, 32ビットのエレメント複数個を連結したストリングの転送、比較、検索を行なうストリング命令(SMOV, SCMP, SSCH, SSTR)を実行する際には二つの演算回路、即ち主演算回路350 と副演算回路351 を並列に動作させて一度に8バイトずつストリングを処理する。従って、1, 2, 4バイトのエレメントではそれぞれ一度に8, 4, 2個を並列に処理する。また、バーストモードによるブロック転送により効率よくバスを使用するため、メモリアクセスは16バイト単位で行なわれる。データロード時には8バイトのデータリードが2回連続して行なわれ、データストア時には16バイトのブロックライト動作が行なわれる。

【0170】ここで、ストリング命令の処理例として、ヌルバイト(=H00)で終わるバイトストリングをSMOV命令で転送する場合を考える。図37の模式図にSMOV命令を処理するマイクロプログラムの第n回目の16バイト処理のループの前半の8バイト処理の状態を示す。

【0171】このSMOV命令では、ソースストリングの先頭アドレスA0はレジスタR0に、デスティネーションストリングの先頭アドレスA1はR1にそれぞれ保持される。まず、2つの8バイトのソースストリングS2nとS2n+1とがメモリからフェッチされてDD部357 のDDROとDDR1とに

保持される。ソースストリングS2n-1は前回のループで既にフェッチ済みであって作業用レジスタに保持されている。A0とA1とは任意のアドレスであり、一般にワード境界のアドレスではない。従って、8バイト境界単位でストリング処理を行なう本発明のデータ処理装置100 では、ソースストリングをデスティネーションストリングの位置に移動するには両ストリングのアドレスの差を8で割った余りであるmバイトだけソースのバイトストリングをシフトする必要がある。そこで、64ビットのパレルシフタによりソースストリングS2n-1を(64-8m)ビット左に、ソースストリングS2nを8mビット右にシフトすることによりT2n-1とT2nとを得る。次に、2つの32ビットALU を並列に動作させてT2n-1とT2nとの論理和として8バイト境界に整置されたデスティネーションストリングD2nを得て、これを作業用レジスタにストアする。デスティネーションストリングD2nを得る際に、ALUでは各バイト毎の論理和出力を並列にゼロ検出することによりヌルストリングの検索を行なう。マイクロプログラムの第n回目の16バイト処理のループの後半の8バイト処理では前半の処理と同様にソースストリングS2nとS2n+1とからデスティネーションストリングD2n+1が得られる。そして、16バイト境界のデータ(D2n+D2n+1)がバーストモードによるブロック転送でメモリにストアされる。

【0172】本発明のデータ処理装置100 は上述の1ループのオペレーションを6クロックサイクルで行なう。従って、ヌルストリングで終了するバイトストリングをSMOV命令で処理する場合、処理速度は16バイト/6クロックサイクル=2.7 バイト/クロックサイクルとなる。

【0173】SMOV命令以外のストリング命令及び上述の条件以外でSMOV命令を処理する際にも、本発明のデータ処理装置100 では1つのマイクロ命令で主演算回路350 と副演算回路351 とを制御することにより、2つの32ビットデータを並列処理する。また、16バイト毎のバーストモードによるブロック転送を利用して効率よくメモリアクセスする。図38に各ストリング命令の処理速度のピーク値の一覧表を示す。なお、ストリング命令ではマイクロプログラムのループ処理以外の前処理と後処理との部分で数クロックサイクルの処理時間を必要とするため、短いストリングに対しては単純な命令を使用する場合より処理時間が長くなることがあるので注意が必要である。

【0174】(4.3.5) 「任意長ビットフィールド命令」
本発明のデータ処理装置100 では、バーストモードによるブロック転送で16バイト単位でメモリアクセスし、整数演算部155 でデータを64ビット単位で処理することにより、BVSCCH, BVMAP, BVCPY, BVPATの任意長ビットフィールド命令をストリング命令と同様に効率よく処理する。

【0175】任意長ビットフィールド命令を処理する際

10

20

30

40

50

にも1つのマイクロ命令で整数演算部155の二つの演算回路、即ち主演算回路350と副演算回路351とを並列に動作させて64ビット単位でオペランドを操作する。この結果、本発明のデータ処理装置100の任意長ビットフィールド命令のクロックサイクル毎の処理性能は従来のデータ処理装置の約2倍となる。図39に本発明のデータ処理装置100の任意長ビットフィールド命令の処理速度のピーク値の一覧表を示す。なお、任意長ビットフィールド命令もストリング命令と同様にマイクロプログラムのループ処理以外の部分に数〜十数クロックサイクルの処理時間を必要とするため、短いビットフィールドでは単純命令より処理時間が長くなる場合がある。

【0176】(4.4)「ジャンプ命令の処理」

(4.4.1)「先行ジャンプ処理の概要」

本発明のデータ処理装置100の命令パイプラインは6段構成であり、命令実行を行なうEステージ404は5番目のステージである。このため、命令実行段階でジャンプが実行された場合にはEステージ404に先行するIFステージ400、Dステージ401、Aステージ402及びFステージ403での処理は全てキャンセルされる。各ステージのステージ処理時間は最小でも1クロックであるため、Eステージ404でジャンプを行なった場合には最小でも4クロックの無駄時間が発生する。

【0177】ジャンプ命令実行によるパイプラインの乱れを少なくするため、従来のデータ処理装置では条件分岐命令の履歴を記憶して履歴に従った分岐予測を行なうことによりパイプラインの2番目のステージである命令デコードステージ(Dステージ)401で条件分岐命令を実行したり、あるいはサブルーチンの戻り先アドレスのコピーをチップ内のPCスタック243に保持してサブルーチンリターン命令を同じくDステージ401で実行する等の先行ジャンプ処理を行なっていた。しかし、本発明のデータ処理装置100では従来のデータ処理装置の構成に加えて、パイプラインの3番目のステージであるオペランドアドレス生成ステージ(Aステージ)402においても先行ジャンプを行なったり、あるいは条件分岐命令の非分岐側と分岐側との2系列の命令コードをフェッチするために命令キュー201を2個にする等により先行ジャンプ処理機構を更に強化している。

【0178】本発明のデータ処理装置100において先行ジャンプ処理を行なう命令と、先行ジャンプを行なうために使用するハードウェアとを図40に一覧表として示す。また、本発明のデータ処理装置100のジャンプ命令の処理を行なうステージの関係を図41のブロック図に示す。

【0179】ジャンプ命令はPC生成部154、アドレス生成部153または整数演算部155のいずれかで処理される。ジャンプ先アドレスはジャンプ先からの命令フェッチのためにJAバス171(ジャンプ先アドレスバス)により命令フェッチ部150へ転送される他、ジャンプ先命令に引き

続く命令のPC値を計算するためにPC生成部154へも転送される。

【0180】PC生成部154はDステージ401で動作し、Dステージ401での先行ジャンプを処理する。アドレス生成部153はAステージ402で動作し、Aステージ402での先行ジャンプを処理する。Aステージ402ではまた、条件分岐命令に対するDステージ401での処理(分岐または非分岐)の反対側の命令列のアドレス計算処理を行なう。整数演算部155はEステージ404で動作し、間違った先行ジャンプからの復帰、EITの起動等のためのジャンプ処理を行なう。

【0181】無条件分岐命令(BRA)に対してEステージ404でジャンプを行なう場合と、Dステージ401で先行ジャンプを行なう場合とのそれぞれのパイプライン中の命令の流れを前者を図42に、後者を図43のタイミングチャートに示す。Eステージ404でジャンプを行なう場合にはジャンプ先命令の実行までに4クロックの無駄時間が発生するが、Dステージ401でジャンプを行なう場合には無駄時間は1クロックに減少する。なお、本発明のデータ処理装置100では先行ジャンプしたBRA命令のA、F、Eステージの処理は行なわれない。

【0182】(4.4.2)「ジャンプ命令の分類」

本発明のデータ処理装置100の命令セットには10種類以上のジャンプ命令が用意されており、その全てのジャンプ命令に対して命令デコードステージ(Dステージ)401またはオペランドアドレス生成ステージ(Aステージ)402で先行ジャンプを行なうことはハードウェアに対する負荷が大きい。また、実行頻度の関係からその必要もない。本発明のデータ処理装置100では実行頻度が低い一部のジャンプ命令に対しては先行ジャンプ処理を行なわない。本発明のデータ処理装置100の各ジャンプ命令に対する先行ジャンプ処理の有無と最小処理時間とを図44に一覧表として示す。

【0183】(4.4.3)「PC相対アドレス及び絶対アドレスのジャンプ処理」

BRA、BSR命令とPC相対アドレスへのJMP、JSR命令とをDステージ401でデコードした場合には、本発明のデータ処理装置100はPC生成部154で分岐先アドレスを計算して先行分岐を行なう。また、分岐予測テーブル450により分岐すると予測したBcc命令、ACB、SCB命令をDステージ401でデコードする場合も、PC生成部154で分岐先アドレスが計算されて先行分岐が行なわれる。絶対アドレスへのJMP、JSR命令をデコードした場合には、命令フィールドから切り出した絶対アドレスへ先行ジャンプする。

【0184】先行分岐または絶対アドレスへの先行ジャンプの際には、分岐変位pcdispとPC値とを加算する分岐先アドレス計算または絶対アドレスabsの切り出し処理はデコード処理と並行して行なわれる。命令デコードを開始する時点ではpcdispフィールドの位置もabsフィー

ルドの位置も不明である。このため、本発明のデータ処理装置100では上述の命令のデコード開始と同時にpcdispフィールド及びabsフィールドの位置を予測してジャンプ先アドレスの生成を開始する。但し、ACB,SCB命令の一部にはpcdispフィールドの予測が困難なために命令デコード後に分岐先アドレスの計算を行なう命令もある。

【0185】PC相対アドレスへのジャンプに対しては、図45に示す4種類のpcdispフィールドについてアドレスが計算される。絶対アドレスへのJMP,JSR命令に対しては、図45に示す2種類のabsフィールドの切り出しと符号拡張処理とが行なわれる。そして、デコード終了時点でジャンプ先アドレスとなり得る計3つの生成値から1つが選択されてIFステージ400へ転送される。

【0186】命令の各pcdispフィールドとabsフィールドとは命令キュー201から48ビットのIIバス169を経由して分岐先アドレス計算部へ転送される。一方、現在デコード中の命令のPC値はPC計算部241からPIバス173を経由して分岐先アドレス計算部へ転送される。命令デコードの結果、判明した先行ジャンプ先アドレスがJAバス171に読み出され、命令フェッチ部150へ転送される。

【0187】Bcc命令の分岐予測は過去1回の実行履歴に従って行なわれる。過去の実行履歴は図46図の模式図に示す1ビット×1Kビットの分岐予測テーブル450に記憶されており、Bcc命令の直前にデコードした命令のPC値の下位11ビット(bit0は常に"0")の値に応じてその内の1つを選択して読み出す。Bcc命令を命令実行ステージ404で実行した場合、分岐予測が間違っていた場合にはBcc命令の直前に実行した命令のPC値の下位11ビット(bit0は常に0)の値に基づいて分岐予測テーブル450が更新される。

【0188】分岐予測テーブル450の初期値は全て「分岐しない」である。分岐予測テーブル450はBMCレジスタとBMPレジスタとにより制御される。BMCレジスタのMビットを"1"にするとリング1~3で分岐予測テーブル450が動作し、Bcc命令に対して動的な分岐予測が行なわれる。BMPレジスタのBPビットに"1"を書き込むと分岐予測テーブル450はパージされ、分岐履歴は全て「分岐しない」となる。

【0189】なお、BMCレジスタの値にかかわらず、DB GACKモードまたは全モードのリング0では分岐予測テーブル450は動作せず、全てのBcc命令は分岐しないと予測される。ACB,SCB命令に対しては常に分岐すると予測して先行分岐処理が行なわれる。これらの命令の分岐予測をソフトウェアで制御することはできない。

【0190】(4.4.4)「先行リターン処理」

サブルーチンからのリターンを行なう2つのサブルーチンリターン命令(RTS,EXITD命令)に対しては、本発明のデータ処理装置100はDステージ401での命令デコード時に先行リターンを行なう。本発明のデータ処理装置

100はサブルーチンからのリターンアドレスを保持する16エントリのスタックメモリを内蔵しており、サブルーチンジャンプ命令(BSR,JSR命令)を実行した場合、外部メモリ上のスタックにプッシュするサブルーチンからのリターンアドレスのコピーをPCスタック243にもプッシュする。

【0191】PCスタック243は図47のブロック図に示すように構成されている。PCスタック243は、最近に実行した16個のサブルーチンジャンプ命令が外部メモリにプッシュしたPC値のコピーを保持する。そして、サブルーチンリターン命令をDステージ401でデコードした際にPCスタック243からリターンアドレスをポップしてそのアドレスへ先行ジャンプする。

【0192】PCスタック243は16エントリのデータ量しか有していないため、サブルーチンのネストレベルが16を超えた場合にはラップアラウンドして古いデータを保持しているエントリにオーバーライトする。また、PCスタック243はDステージ401とEステージ404とからアクセスされるため、それぞれのステージ用のスタックポインタ(DP451,EP452)を備えている。DP451は、サブルーチンリターン命令がDステージ401での先行リターンのためにPCスタック243からPC値をポップする際に使用する。EP452は、サブルーチンジャンプ命令がPC値をPCスタック243にプッシュする際と、サブルーチンリターン命令が先行リターンの検証のためにEステージ404でPCスタック243からPC値をポップするためとに使用する。サブルーチンリターン命令が連続した場合等には、この2つのポインタ451,452の値が一致しない瞬間がある。従って、DP451とEP452とが不一致のままその後の処理が行なわれることを防ぐため、Eステージ404でジャンプが発生した場合にはEP452の内容がDP451にオーバーライトされる。

【0193】PCスタック243からフェッチされた先行リターン先のアドレスはほとんどの場合には正しいが、サブルーチンの実行中にタスクスイッチが発生したり、あるいはサブルーチンのネストレベルが16を超えたりした場合には正しくない場可能性が生じる。このため、先行リターンを行なったサブルーチンリターン命令は外部メモリから真のリターン先アドレスをフェッチし、そのアドレスと先行リターンに使用したアドレスとをPCスタック243からポップして両者をEステージ404で比較する。この場合、2つのアドレスが異なっていれば外部メモリからフェッチした真のリターン先アドレスへジャンプする。

【0194】RTS命令で先行リターンを行なった際のパイプライン中の命令の流れを図48及び図49のタイミングチャートに示す。図48のタイミングチャートに示されているように、先行リターンアドレスが正しい場合には、

RTS命令の実行からリターン先命令であるADD #1,@(-76,FP)命令を実行するまでの無駄時間は1クロックとな

る。一方、図49のタイミングチャートに示されているように、先行リターンアドレスが間違っている場合には RTS命令が E ステージ404 でジャンプが行なわれるため、4クロックの無駄時間が発生する。PCスタック243 はDB GACKモードでは動作しない。その他のモードではPCスタック243 は必ず動作し、ソフトウェアでPCスタック243 の動作を制御したり、あるいはPCスタック243 の内容を書き換えたりは出来ない。

【0195】(4.4.5) 「命令プリフェッチキュー」

Bcc命令及びACB、SCB命令は条件分岐命令であるため、本発明のデータ処理装置100 の先行分岐が必ずしも正しいとは限らない。条件分岐命令が分岐すると予測して先行分岐を行なったが実際には分岐しなかった場合は、E ステージ404 でその命令の直後の命令に分岐して正しいシーケンスに復帰する。条件分岐命令が分岐しないと予測したが実際には分岐した場合は、命令実行段階で分岐先命令へ分岐する。

【0196】本発明のデータ処理装置100 では条件分岐命令の実行速度を向上させる目的で、非分岐側の命令列をプリフェッチする命令キュー A 201A と分岐側の命令列をプリフェッチする命令キュー B 201B との計2つの命令プリフェッチキューを有する。図50のブロック図に示す如く、各命令プリフェッチキュー201A、201Bはいずれも16バイトの容量を有し、命令キャッシュ200 からは8バイト（16バイト境界内の任意の8バイトを一度にフェッチ可能）単位で命令をフェッチし、外部からは4バイト（整置された4バイトに限る）単位で命令をフェッチし、命令デコード部151 がデコードする命令コード量にに応じて2～8バイトの単位で命令を出力する。

【0197】命令キュー A 201A(または命令キュー B 201B)から出力された条件分岐命令を D ステージ401 でデコードして先行分岐を行なった場合、命令キュー A 201A(または命令キュー B 201B)の内容はそのままにしておき、命令キュー B 201B(または命令キュー A 201A)をクリアして分岐先命令の命令をプリフェッチする。命令デコード部151 は命令キュー B 201B(または命令キュー A 201A)から出力される命令コードをデコードする。なお、分岐先命令の先頭の16バイト境界内の8バイトの命令コードは命令キュー B 201B(または命令キュー A 201A)をバイパスして命令デコード部151 へも転送される。

【0198】先行分岐を行なった条件分岐命令を A ステージ402 で処理する際に分岐予測が誤っていた場合、PC生成部154 を初期化するためにオペランドアドレス生成部153 では非分岐先アドレス（次命令のアドレス）が計算される。先行分岐を行なった条件分岐命令の分岐条件を命令実行ステージ404 で評価して分岐しない条件分岐命令であると判断した場合、本発明のデータ処理装置100 は命令キュー B 201B(または命令キュー A 201A)と D、A、F ステージで処理中の命令とをキャンセルして命令処理の流れをその命令の直後の命令列に復帰し、命

令デコード部151が命令キュー A 201A(または命令キュー B 201B)から出力される命令コードをデコードする。このため、分岐予測結果が正しくなくて間違った先行ジャンプを行なった場合でも、命令処理の流れは命令キュー A 201A(または命令キュー B 201B)にある元の命令列に速やかに復帰する。

【0199】図51のタイミングチャートに、BNE命令に対して間違った分岐予測に従って先行分岐を行なった場合のパイプライン中の命令の流れを示す。BNE命令が D ステージ401 で先行分岐を行ない、BNE命令に引き続く「MOV @R6,R7」以下の命令列を命令キュー A 201A が保持したまま、分岐先命令であるラベルL21 から始まる「MOV @R6,R4」以下の命令列をもう一つの命令キュー B 201B がフェッチし、D ステージ401 以後のパイプライン処理が開始される。BNE命令を E ステージ404 で実行して分岐予測が間違っていたことが判明すると、分岐先命令の処理がキャンセルされて非分岐側の「MOV @R6,R7」以下の命令列の D ステージ401 以後のパイプライン処理が開始される。非分岐側命令列は BNE命令で先行分岐を行なった後も命令キュー A 201A に保持されているため、BNE命令が E ステージ404 で非分岐側命令列に復帰するための処理を行なった場合に新たに命令をフェッチする必要がない。

【0200】命令キュー A 201A(または命令キュー B 201B)から出力された条件分岐命令を D ステージ401 でデコードした結果、先行分岐を行なわない場合、命令デコード部151 は引続き命令キュー A 201A(または命令キュー B 201B)の出力をデコードする。命令キュー B 201B(または命令キュー A 201A)は分岐予測が誤っていた場合に備えて D ステージ401 で計算した分岐先アドレスに従って命令をフェッチする。先行分岐を行なわなかった条件分岐命令を A ステージ402 で処理する際に分岐予測が誤っていた場合には PC生成部154 を初期化する必要があるため、オペランドアドレス生成部153 では分岐先アドレスを計算する。先行分岐を行なわなかった条件分岐命令の条件コードを命令実行ステージ404 で評価した結果、分岐する条件分岐命令であると判断した場合、本発明のデータ処理装置100 は命令キュー A 201A(または命令キュー B 201B)と D、A、F ステージで処理中の命令とをキャンセルして命令処理の流れを分岐先の命令列へ移行する。また、命令デコード部151 は命令キュー B 201B(または命令キュー A 201A)から出力される命令コードをデコードする。このため、分岐予測結果が正しくなくて先行ジャンプを行なわなかった場合でも、命令処理の流れは命令キュー B 201B(または命令キュー A 201A)にある分岐先の命令列に速やかに移行する。

【0201】図52のタイミングチャートに、BNE命令に対して間違った分岐予測に従って先行分岐を行なわなかった場合のパイプライン中の命令の流れを示す。BNE命令は D ステージ401 で先行分岐を行なわないためパイプ

ラインは BNE命令に引き続く非分岐側命令列の処理を継続する。しかし、BNE命令が E ステージ404 で分岐することに備えて BNE命令の分岐先アドレスが D ステージ401 で計算され、命令フェッチ部150 はラベルL21 から始まる「MOV @R6,R4」以下の命令列を非分岐側命令を保持する命令キュー A 201A とは異なるもう一つの命令キュー B 201B にフェッチする。BNE命令の条件コードを命令実行ステージ404 で評価した結果、分岐する条件分岐命令であることが判明すると、非分岐先命令の処理がキャンセルされて分岐側の「MOV @R6,R4」以下の命令列の D

【0202】2つの条件分岐命令を連続して処理したり、ほとんど隔たりなく連続して処理する場合、先行する条件分岐命令が命令実行ステージ404 に到達する前に後続の条件分岐命令の分岐先から命令をフェッチする要求が出される場合がある。この場合、先行する条件分岐命令が実行されて分岐、非分岐が決定するまで2つの命令キューはいずれも内容を保持する必要がある、後続の条件分岐命令の分岐先命令をフェッチすることはできない。本発明のデータ処理装置100 ではこのため、後続の条件分岐命令の分岐先アドレスを登録する専用の分岐先アドレスレジスタ202 を命令フェッチ部150 に備えている。先行する条件分岐命令が実行され、後続する条件分岐命令の処理が有効となった段階で、命令フェッチ部150 は分岐先アドレスレジスタ202 のアドレスからフリーになった命令キューに命令をプリフェッチする。

【0203】サブルーチンリターン命令及び無条件ジャンプ命令に対しては、命令キューは1つしか使用されない。無条件ジャンプ命令に対して D ステージ401 または A ステージ402 で先行ジャンプを行なう場合、現在使用中の命令キューの内容がクリアされてその命令キューにジャンプ先の命令がプリフェッチされる。この際、もう一方の命令キューは先行する条件ジャンプ命令の分岐予測と反対側の命令を保持したままになる。また、無条件ジャンプを行なう命令をデコードした場合、本発明のデータ処理装置100 の命令デコード部151 はジャンプが実行されるまでの間、後続命令のデコードを休止する。先行リターンが間違っていたり、EITが起動されたりすることにより E ステージ404 でジャンプが発生した場合は2つの命令キューは共にクリアされる。

【0204】(4.4.6) 「各ジャンプ命令の動作」
以下に、本発明のデータ処理装置100 が備える各ジャンプ命令の処理について説明する。

【0205】1) BRA命令

BRA命令は D ステージ401 で必ず先行ジャンプを行なう。PC生成部154 では BRA命令のデコードと並列して B

RA命令の分岐先アドレスが計算され、BRA命令のデコード直後に先行分岐を行なう。従って、BRA命令の処理時間は命令のフェッチとデコードとに要する時間の計2クロックサイクルとなる。BRA命令では現在使用中の命令キューがクリアされ、分岐先アドレスから新たに命令がフェッチされる。BRA命令は A ステージ402 以降では処理されず、D ステージ401 での処理完了後にパイプライン中で消滅する。但し、セルフデバッグ中または DBGデバッグ中はPC値を転送するためにパイプライン中から消滅せずに A ステージ402 以降も処理される。

【0206】2) JMP命令

JMP命令はジャンプ先アドレッシングモードに依存して D ステージ401 または A ステージ402 で先行ジャンプを行なう。絶対アドレスモードまたは32ビットディスプレイメント付きPC相対モードのJMP命令は D ステージ401 で先行ジャンプを行なう。D ステージ401 では JMP命令に対して命令デコードと並列にジャンプ先アドレスの計算が行なわれるため、D ステージ401 で先行ジャンプを行なう JMP命令の処理時間は BRA命令と同じく2クロックサイクルとなる。16ビットディスプレイメント付きPC相対モードのJMP命令は A ステージ402 で先行ジャンプを行なうため、処理時間が3クロックサイクルとなる。

【0207】絶対アドレスモード及びPC相対モード以外のアドレッシングモードの JMP命令に対しては A ステージ402 で必ず先行ジャンプを行なう。A ステージ402 の処理時間はアドレッシングモードに依存して異なるため、A ステージ先行ジャンプを行なう JMP命令の処理時間は最小3クロックで、アドレッシングモードが複雑になればその分増加する。

【0208】3) Bcc命令

Bcc命令は過去1回の実行履歴による分岐予測に従って D ステージ401 で先行ジャンプを行なう場合と、常に先行ジャンプを行なわない場合とがある。実行履歴は1Kバイトのテーブルに記憶されており、Bcc命令の直前にデコードした命令の下位11ビット（最下位ビットは常にゼロ）の値に従って実行履歴が参照されて Bcc命令の分岐予測が行なわれる。

【0209】先行ジャンプの有無に関係なくPC生成部154 は Bcc命令のデコードと並列に D ステージ401 で分岐先アドレスを計算し、命令フェッチ部150 へ転送する。分岐すると予測した場合は非分岐先命令コードを命令キュー201 に保持したまま、もう一方の命令キューに分岐先命令をフェッチしてデコード処理が開始される。従って、分岐予測が正しい場合の先行ジャンプを行なう Bcc命令の処理時間は2クロックサイクルとなる。分岐しないと予測した場合は非分岐先命令コードが現在使用中の命令キューから出力されつつもう一方の命令キューに分岐先命令コードがフェッチされる。従って、分岐予測が正しい場合の先行ジャンプを行なわない Bcc命令の

処理時間は1クロックサイクルとなる。

【0210】Bcc命令の真の分岐条件はEステージ404で評価され、分岐予測が正しかった場合はパイプラインを乱すことなく処理を終了するが、分岐予測が誤っていた場合には待機中の命令キューに保持されている非分岐先命令または分岐先命令に処理が切り替えられる。従って、分岐予測が正しくない場合のBcc命令の処理時間は分岐予測結果とは無関係で、4クロックサイクルとなる。なお、分岐予測が誤っていた場合にはBcc命令の直前にEステージ404で実行した命令の下位11ビット（最下位ビットは常にゼロ）の値に基づいて分岐履歴の更新も行なわれる。Bcc命令をAステージ402で処理する場合は分岐予測が誤っていたことに備えて分岐予測と反対側の命令アドレスを計算し、分岐予測が外れた場合にEステージジャンプを発行してPC生成部154を初期化するために使用する。

【0211】4)ACB, SCB命令

ACB命令とSCB命令とはBMCレジスタのMビット、実行したリングあるいは分岐履歴には関係なく、常に分岐すると予測してDステージ401で先行ジャンプを行なう条件分岐命令である。本発明のデータ処理装置100ではACB, SCB命令の先行ジャンプをディセーブルすることはできない。ACB, SCB命令にはそれぞれ複数の命令フォーマットがあり、命令フォーマットとpcdisp長とに依存して先行ジャンプ処理に必要な時間が異なる。

【0212】Q, Rフォーマットでpcdispが8ビットである場合はDステージ401でACB, SCB命令のデコードと並列して分岐先アドレスの計算が行なわれ、非分岐先命令を一方の命令キューに保持したままもう一方の命令キューに分岐先命令がフェッチされてデコード処理が開始する。従って、分岐する場合の命令処理時間は2クロックサイクルである。Eステージ404で分岐条件が評価され、分岐しないと判明した場合は命令キューに保持されている非分岐先命令のデコードを再開する処理が行なわれるため、命令の処理時間は7クロックサイクルとなる。

【0213】G, Eフォーマットでpcdispが8ビットである場合はDステージ401で3クロックサイクルを要してACB, SCB命令をデコードした後に分岐先アドレスの計算が行なわれ、非分岐先命令を一方の命令キューに保持したままもう一方の命令キューに分岐先命令がフェッチされてデコード処理を開始する。従って、分岐する場合の命令処理時間は4クロックサイクルである。Eステージ404で分岐条件が評価され、分岐しないと判明した場合は命令キューに保持されている非分岐先命令のデコードを再開する処理が行なわれるため、命令の処理時間は8クロックサイクルとなる。

【0214】pcdispが8ビット以外である場合はフォーマットによらず命令の処理時間は同じである。この場合もDステージ401でACB, SCB命令がデコードされ、分岐

先アドレスが計算されて非分岐先命令を一方の命令キューに保持したままもう一方の命令キューに分岐先命令がフェッチされてデコード処理を開始する。分岐する場合の命令処理時間は3クロックサイクルである。Eステージ404で分岐条件が評価され、分岐しないと判明した場合は命令キューに保持されている非分岐先命令のデコードを再開する処理が行なわれるため、命令の処理時間は8クロックサイクルとなる。

【0215】5)BSR命令

BSR命令はジャンプ先のアドレッシングモードに依存してDステージ401でBSR命令のデコードと並列して分岐先アドレスが必ず計算される。Aステージ402ではサブルーチンからの復帰先アドレスであるBSR命令の次の命令のアドレスが計算される。Eステージ404とSステージ405とはその復帰先アドレスをそれぞれPCスタック243とメモリのスタックトップとへプッシュする。従って、BSR命令の処理時間は2クロックサイクルである。

【0216】6)JSR命令

JSR命令はジャンプ先のアドレッシングモードに依存してDステージ401またはAステージ402で先行ジャンプ処理を行なう。絶対アドレスモードまたは32ビットディスプレイメント付きPC相対モードのJSR命令はDステージ401で先行ジャンプを行なう。Dステージ401ではJSR命令に対して命令デコードと並列にジャンプ先アドレスの計算が行なわれるため、Dステージ401で先行ジャンプを行なうJSR命令の処理時間はBSR命令と同じく2クロックサイクルとなる。16ビットディスプレイメント付きPC相対モードのJSR命令はAステージ402で先行ジャンプを行なうため、処理時間が3クロックサイクルとなる。サブルーチンからの復帰先アドレスはAステージ402でアドレス加算器を使用してジャンプ先アドレスの計算の次のクロックサイクルで計算され、E, SステージでそれぞれPCスタック243とメモリのスタックトップとにその復帰先アドレスがプッシュされる。

【0217】絶対アドレスモード及びPC相対モード以外のアドレッシングモードのJSR命令に対してはAステージ402で必ず先行ジャンプを行なう。Aステージ402の処理時間はアドレッシングモードに依存して異なるため、Aステージ先行ジャンプを行なうJSR命令の処理時間は最小3クロックで、アドレッシングモードが複雑になればそれに対応して増加する。サブルーチンからの復帰先アドレスはDステージ401でPC生成部154を用いて計算され、E, SステージでそれぞれPCスタック243とメモリのスタックトップとにその復帰先アドレスがプッシュされる。

【0218】7)RTS命令

RTS命令はPCスタック243からリターン先アドレスをフェッチすることによりDステージ401で先行ジャンプを行なう。命令デコード部151でRTS命令をデコードした場合はPC生成部154のPCスタック243からリターン先ア

ドレスがポップされ、Dステージ401 でそのアドレスへプリリターンする。Aステージ402 ではスタックポインタがプリデクリメントされ、Fステージ403 でメモリのスタックから真のリターン先PCがポップされる。Eステージ404 では再びPCスタック243からプリリターンに使用したアドレスがポップされ、Fステージ403 でメモリからポップした真のリターン先アドレスと比較される。プリリターンアドレスと真のリターン先アドレスとが一致すればプリリターンが正しく行なわれたことになり、Eステージ404 でのジャンプ動作は行なわれない。一致しなければプリリターンが正しくないと判断され、Eステージ404 で真のリターン先アドレスへジャンプする。従って、RTS命令の処理時間はプリリターンが正しい場合2クロックサイクル、プリリターンが正しくない場合6クロックサイクルとなる。

【0219】8) EXITD実

EXITD命令は RTS命令と同じくPCスタック243 からリターン先アドレスをフェッチすることによりDステージ401 で先行ジャンプを行なう。命令デコード部151 で EXITD命令をデコードした場合はPC生成部154 のPCスタック243 からリターン先アドレスがポップされ、Dステージ401 でそのアドレスへプリリターンする。EXITD命令はAステージ402 で複数のステップコードに分解されて処理される。最後のステップコードに対応するFステージ処理でメモリのスタックから真のリターン先PCがポップされる。Eステージ404 では再びPCスタック243 からプリリターンに使用したアドレスがPCスタック243 からポップされ、Fステージ403でメモリからポップした真のリターン先アドレスと比較される。プリリターンアドレスと真のリターン先アドレスとが一致すればプリリターンは正しく行なわれたことになり、Eステージ404 でのジャンプ動作は行なわれない。一致しなければプリリターンが正しくないと判断され、Eステージ404 で真のリターン先アドレスへジャンプする。EXITD命令の処理時間は復帰するレジスタ数に依存して異なるが、リターン動作のみに要する時間はプリリターンが正しい場合には2クロックサイクルに、プリリターンが正しくない場合には6クロックサイクルにそれぞれなる。

【0220】(4.5) 「パイプラインインターロック」

本発明のデータ処理装置100 の命令パイプラインでは、先行命令のデータの書き込みと後続命令のデータの読み出しの間に RAWデータハザード(Read-After-Write data hazard)が発生することを防止する目的で、ハードウェアによるパイプラインインターロック機構を備えている。RAWデータハザードは先行命令の演算結果の書き込みとオペランドのアドレス生成のための後続命令のデータの読み出しとが近接した命令間で行なわれる場合、及び先行命令の演算結果のメモリへの書き込みと後続命令のメモリデータの読み出しとが近接した命令間で行なわれる場合に発生する可能性がある。本発明のデータ処理

装置100 ではこの二種類の RAWデータハザードをスコアボードレジスタ480 及びタグアドレス付きオペランドプリフェッチキュー291 により回避している。以下、上述の二種類の RAWデータハザードの回避を目的とした構成とそのためにパイプラインインターロックで無駄になる処理時間について説明する。

【0221】(4.5.1) 「オペランドアドレスの生成」

図53のブロック図に本発明のデータ処理装置100 のオペランドアドレス生成機構の構成例を示す。命令パイプライン中でオペランドのアドレスを生成するAステージ402 はレジスタへの演算結果の書き込みを行なうEステージ404 よりも2ステージ前段にあり、またメモリへの演算結果の書き込みを行なうSステージ405 よりも3ステージ前段にある。従って、先行命令が演算結果をレジスタまたはメモリに書き込む可能性がある場合、後続命令は同じレジスタまたはメモリ領域からデータを読み出すことができない。

【0222】本発明のデータ処理装置100 ではオペランドアドレス生成時に発生するこの RAWデータハザードをハードウェアが検知してパイプラインをインターロックする。先行命令のオペランド書き込み場所はその命令がAステージ402 で処理された際に図54の模式図に示すスコアボードレジスタ480 に登録される。後続命令のAステージ処理を行なう場合、ハードウェアがこのスコアボードレジスタ480 をチェックしてオペランドのアドレス生成に使用するデータと先行命令の書き込みオペランドとで RAWデータハザードが発生するか否かを調べる。そして、RAWデータハザードが検知された場合には、後続命令の処理は先行命令のオペランド書き込みが終了するまでAステージ402 でストールされる。

【0223】スコアボードレジスタ480 は16個の汎用レジスタ20G のそれぞれについてデータの書き込みの有無をチェックし、データの書き込みがある場合はAステージ402 で対応する位置に書き込み予約ビット"1"が登録される。書き込み予約ビットはパイプライン中の命令と同期してスコアボードレジスタ480 中をシフトする。汎用レジスタ20G へのオペランド書き込みはEステージ404 で行なわれるため、レジスタの登録ビットはEステージ404 で命令が実行された後にスコアボードレジスタ480 からシフトアウトされる。図54では説明を簡単化するためにF、Eステージとそれぞれに1つのスコアボードレジスタ480 を対応させた模式図を示しているが、実際のスコアボードレジスタ480 はキューイングするRコード226とパイプライン間の作業用ラッチともに対応して存在しており、合計4つのスコアボードレジスタが存在する。

【0224】後続命令がAステージ402 でオペランドのアドレス生成を行なうために汎用レジスタ20G またはメモリからデータを読み出す際には、RAWデータハザードチェック回路481 が読み出しリソースに書き込み予約が

行なわれているか否かを調べる。RAWデータハザードが検出された場合には、先行命令の処理が進み、書き込み予約ビットがスコアボードレジスタ480 からシフトアウトされるまで後続命令はAステージ402 でストールする。図55のタイミングチャートに RAWデータハザードによるパイプラインインターロックによりAステージ402 で命令がストールする場合の、また図56のタイミングチャートにコンパイラがパイプラインストールを回避するために命令の順序を変更した場合のパイプライン処理の流れをそれぞれ示す。

【0225】図55では「ADD R3,R2」命令の演算結果を格納するデスティネーションレジスタと引き続く「ADD #1,@(-4,R2)」命令のオペランドアドレス計算に使用するベースレジスタとが同一であるために RAWデータハザードが検出され、パイプラインがインターロックされて「ADD #1,@(-4,R2)」命令が2クロックサイクルに亘ってAステージ402 でストールしている。一方、図56では命令の実行順序を入れ替えることにより、RAWデータハザードを起こす上述の2命令の間にこれらの命令の演算とは独立関係の5つの命令を挿入してパイプラインイン

ターロックによるストールを排除し、命令の実行に要する時間を2クロックサイクル少なくすることに成功している。コンパイラは図55に示すオペランドアドレス生成に伴うパイプラインインターロックがなるべく発生しないように命令を配置すべきである。

【0226】本発明のデータ処理装置100 の命令パイプラインは上述の RAWデータハザードの問題と同じく、スタックプッシュまたはスタックポップによるスタックポインタ値の変更と後続命令のオペランドアドレス生成とに関してもやはり RAWデータハザードの問題がある。しかも、スタックプッシュまたはスタックポップアドレッシングはアークギュメントのプッシュまたはレジスタの回避復帰のために連続して使用される可能性が高い。従って、本発明のデータ処理装置100 ではスタックプッシュとスタックポップとに伴う RAWデータハザードを回避するため、図57のブロック図に示すように、Aステージ402 からEステージ404 までの各パイプラインステージに作業用スタックポインタを備えている。Aステージ402 の作業用スタックポインタとEステージ404 の作業用スタックポインタとは1, 2, 4, 8のインクリメント／デクリメントが可能であり、スタックプッシュまたはスタックポップにより独立して作業用SP値を変更する。

【0227】Aステージ402 で計算されたスタックポインタ値はパイプライン中の命令の流れに従って図57のFステージ作業用SP 501, Eステージ作業用SP 502の順に転送され、命令終了段階で現SMRNG のSP(SPI, SP0, SP1, SP2, SP3の1つ)に格納される。従って、後続の命令はAステージ402 でオペランドのアドレス生成を行なう際に、即座に最新のスタックポインタ値をAステージ作業用SP 500から読み出すことが可能であるので、パイ

プラインストールは発生しない。

【0228】図58に先行命令のオペランドと後続命令のアドレッシングモードとの組み合わせと、各組み合わせに対するパイプラインストールの有無の一覧を示す。なお、レジスタ相対アドレスへのジャンプ等のように、汎用アドレッシングモードでジャンプ先アドレスを指定するジャンプ命令のジャンプ先アドレスの計算においてもオペランドアドレス計算時と同様に RAWデータハザードが発生する可能性がある。本発明のデータ処理装置100 では、ジャンプ命令のジャンプ先アドレスの計算においてもオペランドアドレスの計算と同じパイプラインインターロック機構が動作するため、間違ったアドレス計算が行なわれることはない。

【0229】(4.5.2)「メモリオペランドのリードライト」

本発明のデータ処理装置100 の命令パイプラインでは、命令の実行ステージであるEステージ404 の前段にメモリオペランドのフェッチを行なうFステージ403 を備え、先行命令の命令実行中に後続命令のメモリオペランドをプリフェッチする。プリフェッチしたオペランドは図59の模式図に示す2エントリのオペランドプリフェッチキュー291 に格納される。メモリオペランドのプリフェッチは先行命令の種類とは無関係に行なわれるため(但し、先行命令が条件分岐命令Bccまたは条件トラップ命令であってデータキャッシュミスの際には後続命令のメモリオペランドのプリフェッチは行なわれない)

、先行命令がメモリにデータをストアする場合にはプリフェッチデータとストアデータとに重複する部分があるか否かをチェックする必要がある。

【0230】このため、本発明のデータ処理装置100 ではプリフェッチオペランドをアドレスタグ付きで整置された8バイト単位で保持し、先行命令のストア動作が行なわれた際のデータの重複をチェックする。Fステージ403 でフェッチした8バイト境界毎のオペランドはそのアドレスと共にプリフェッチキューに保持され、先行命令がストアバッファ(SBDATA)292 にメモリオペランドを書き込む際に、書き込みデータのアドレスがチェックされた上で、重複したバイトのデータがDDバス164 から上書きされる。

【0231】オペランドがストアされる場合、Eステージ404 でデータキャッシュのタグをアクセスしてキャッシュヒット／ミスGA調べられる。キャッシュヒットの場合、Sステージ405 でデータキャッシュのデータ部をアクセスしてヒットしたエントリがオーバーライトされると同時に図60の模式図に示すストアバッファ(SBDATA)292 にストアアドレス、ストアデータ、PC値、ストア情報が格納される。従って、1回のストア動作でデータキャッシュが2クロックサイクル(E, Sステージ)の間ビジーとなり、その直後にメモリからオペランドをフェッチする命令がある場合、その命令は2クロックサイクル

ストールすることになる。

【0232】メモリへのデータストアが連続する場合は、先行命令のSステージでの動作（データキャッシュのデータ部アクセス）と後続命令のEステージでの動作（データキャッシュのタグ部アクセス）とがオーバーラップして実行されるため、ストアバッファ(SBDATA)292に空きがある限りは1クロックサイクル毎にストア動作が行なわれる。ストアバッファ(SBDATA)292では8バイト境界毎のストアオペランドとその物理アドレスをそのストアオペレーションを行なった命令のPC値とストアに関連する情報と共に保持する。本発明のデータ処理装置100ではメモリデータをストアする場合、データキャッシュの書き換え動作を行なうと同時に2エントリのストアバッファ(SBDATA)292にデータを登録する。従って、ストアバッファ(SBDATA)292のデータと重複するアドレスのデータキャッシュの内容は既書き変わっている。また、ストアバッファ(SBDATA)292からメモリへの書き込みは他の全てのメモリアクセスより高い優先順序で行なわれるため、ストアバッファ(SBDATA)292のデータと後続命令のフェッチデータとの間でRAWデータハザードが発生することはない。

【0233】図61のタイミングチャートにメモリオペランド間のRAWデータハザードを避けるためパイプライン中で後続命令がストールする場合の、また図62のタイミングチャートにコンパイラが命令の順序を入れ換えてこのストールを回避した場合のパイプライン処理の流れをそれぞれ示す。

【0234】図61では、先行命令「MUL #10,@(RO)」がメモリヘデータストアを行ない、後続命令「MOV @(8,FP),RO」がメモリからデータフェッチする場合、2つの命令間でのRAWデータハザードの発生を回避するために、先行命令「MUL #10,@(RO)」のストアバッファ(SBDATA)292へのストア処理が終了するまでは後続命令「MOV @(8,FP),RO」はオペランドプリフェッチキュー291の内容を使用することができず、パイプライン中でストールする。また、次の命令「MOV @(12,FP),R1」は、先行命令によりデータキャッシュが使用されているため、オペランドフェッチができず、Fステージ403の開始までに2クロックサイクルの間ストールしている。一方、図62では命令の順序を入れ替えることにより、メモリオペランド間のRAWデータハザードによるパイプラインインターロックによる命令のストールを回避し、処理時間を1クロック削減している。

【0235】メモリへのオペランドをストアする命令とメモリからのオペランドとをフェッチする命令とが連続した場合、後続命令がストールする時間は先行命令のストアデータが8バイトの境界をクロスするか否かにも依存する。メモリオペランド間のRAWデータハザードによる本発明のデータ処理装置100のストール時間を図63に一覧表として示す。

【0236】(4. 5. 3) 「ジャンプ命令及び後続命令の前処理」

本発明のデータ処理装置100の命令パイプラインでは、命令の実行ステージであるEステージ404の前段に命令のプリフェッチを行なうIFステージ400及びメモリオペランドのフェッチを行なうFステージ403を備え、先行命令の命令実行中に後続命令をプリフェッチし、更にその命令のメモリオペランドをプリフェッチする場合もある。特に本発明のデータ処理装置100では、命令デコード段階で分岐予測により先行分岐を行なうため、後続命令は非分岐側になる場合と分岐側になる場合との両方の可能性がある。ここではそのような両方の命令の処理を考える。

【0237】プリフェッチは全てリードオペレーションであり、I/O領域以外から行なわれる限りはメモリの内容を破壊することはない。しかし、プリフェッチはプログラマから見れば本来フェッチすべきでないメモリアドレスに対するリードオペレーションを引き起こし、そのアドレスに対してバスから何の応答もなくハングアップしたり、ウォッチドグタイマが応答するまでの長い期間に亘ってバスサイクルを専有して全体の性能を低下させる等の問題がある。特に条件ジャンプを行なう命令(Bcc, ACB, SCB, TRAP)の実行直後の命令が不合理なアドレスをアクセスする場合に問題となる。

【0238】本発明のデータ処理装置100ではこのような問題に対して、以下に示す手法で非合理的アドレスへのリードオペレーションを回避している。但し、以下の手法では命令プリフェッチの多く（非分岐側アドレス、PC相対ジャンプ先アドレス）は回避されないため、命令のプリフェッチが非合理的アドレスに対して行なわれないように注意する必要がある。

【0239】1)無条件ジャンプ命令に引き続く命令（非ジャンプ側命令）の処理を行なわない。但し、非ジャンプ側命令のフェッチは行なわれる（図64の命令列の模式図参照）。

2)条件ジャンプを行なう命令(Bcc, ACB, SCB, TRAP)ではジャンプ、非ジャンプが確定するまでその後全ての命令のオペランドプリフェッチを外部のメモリからは行なわない（図65の命令列の模式図参照）。

3)条件ジャンプ行なう命令(Bcc, ACB, SCB, TRAP)ではジャンプ、非ジャンプが確定するまでその後のジャンプ命令では絶対アドレス、PC相対以外のジャンプ先アドレス計算処理を行なわない（図66の命令列の模式図参照）。

。1)は本来実行しない命令の処理を中止するのみで1)が原因となるパイプラインインターロックによる性能低下を防ぐには無条件ジャンプ自体を削減するしかない。

2)は条件ジャンプを行なう命令の直後に非キャッシュ領域からのロード命令等を実行する場合に問題になる。但し、内蔵データキャッシュがヒットする場合のリードアクセスまたは内蔵データRAMへのリードアクセスは先行

する条件ジャンプを行なう命令の条件決定前に行なうため問題はない。3)の頻度は非常に小さいと考えられるが、この場合にはパイプラインインターロックにより2クロックサイクルの無駄時間が発生する。3)の場合、本発明のデータ処理装置100は先行する条件ジャンプを行なう命令がEステージ404で処理されるまで、後続するジャンプ命令がAステージ402でストールする。

【0240】(5)「外部アクセス動作」

(5.1)「入出力信号線」

図67の模式図に本発明のデータ処理装置100の入出力信号を示す。本発明のデータ処理装置100は電源Vccと接地GND、32本のデータピンと32本のアドレスピン、入力クロックCLKの他に種々の制御信号を入出力する。なお、図67及び以下の説明において「#」は負論理信号であることを意味している。

【0241】CLK 600は外部入力クロックであり、本発明のデータ処理装置100の動作クロックと同一周波数のクロックである。#RESET 601は本発明のデータ処理装置100を初期化する信号である。アドレスストローブ#AS 602はアドレスピンへ出力されたデータアドレスが有効であることを示す。リードライト信号R/#W 603はデータピンでのバスサイクルが入力か出力かを区別する。

【0242】データストローブ信号#DS 604は本発明のデータ処理装置100がデータ入力準備を完了したこと、または本発明のデータ処理装置100からデータが出力されたことを示す。#DC 605は本発明のデータ処理装置100にデータアクセスサイクルを終了してもよいことを通知する信号である。2ビットの信号BAT1:2 606はアドレスピンへ出力されるアドレスの種類を示す。3ビットのバイトコントロール信号BC0:3 607は4バイト幅のデータバス102の各バイトのデータが有効であるか無効であるかを示す。

【0243】ホールドリクエスト信号#HREQ 608は本発明のデータ処理装置100にバス権を要求する信号である。#HACK 609は本発明のデータ処理装置100がホールドリクエスト信号#HREQ 608を受け付けてバス権を他のデバイスに渡したことを示す信号である。3ビットのIRLO:2 610は外部割込み要求信号である。#IACK 611は本発明のデータ処理装置100が外部割込み要求信号IRLO:2 610に応じて外部割込みを受け付け、割込みベクトルアクセスサイクルを行なっていることを示す信号である。ブロック転送リクエスト信号#BL 612はバスサイクルを4回のバースト転送で行なうことを要求する信号である。信号#BURST 613はバースト転送を受け付けたことを示す信号である。

【0244】(5.2)「外部デバイスのアクセス」

図1に示す本発明のデータ処理装置100を用いたシステムの例では、本発明のデータ処理装置100と主記憶装置103、周辺回路105及びASIC 104とがデータピンにつながるデータバス102、アドレスピンにつながるアドレスバス

101の他、各信号BAT1:2 606, #BC0:3 607, #AS 602, R/#W 603, #DS 604, #DC 605, #BL 612, #BURST 613でも接続されている。CLK 600はシステム全体に供給されシステムの基本タイミングを決定する。

【0245】本発明のデータ処理装置100では信号#BS 614によってバスサイクルが起動する。本発明のデータ処理装置100の基本的なバスオペレーションを図68乃至図73のタイミングチャートに示す。なお、図68及び図69のタイミングチャートが単一転送の例を示し、図70乃至図73のタイミングチャートがバーストモードのブロック転送の例である。

【0246】バスサイクルの開始、即ちクロックCLK 600の立ち上りに同期して信号#BS 614がアサートされ、以下の信号R/#W 603, BAT1:2 606, #BC0:3 607, A00:31 616, #BC0:3 607, #BL 612にアクセスに関する情報が出力される。これらに続いて信号#AS 602がCLK 600の立ち上りに同期してアサートされ、信号#DS 604が次のCLK 600の立ち上りに同期してアサートされる。信号#AS 602のローレベル出力は上述の各信号が有効なことを示しており、外部回路は信号#AS 602の立ち下りでこれらの信号を取り込むことが可能になる。ライトサイクルであれば、信号#DS 604のアサートと同時にデータ信号D00:31 615へデータが出力される。信号#BS 614はバスサイクル中の開始1クロックサイクル後のCLK 600の立ち上りでネゲートされ、信号#AS 602は次のCLK 600の立ち下がりネゲートされる。2クロックサイクル後のCLK 600の立ち上りからクロックサイクル毎に信号#DC 605がサンプリングされ、信号#DC 605にローレベルが入力されたCLK 600の立ち上がり同期して信号#DS 604がネゲートされる。アクセスに関する各信号は信号#DS 604がネゲートされるまで有効である。なお、各図中の○印はサンプリングタイミングを示す。

【0247】図68のタイミングチャートは単一転送によるゼロウェイト及び1ウェイトのリードオペレーションのタイミングを示す。図69のタイミングチャートは単一転送によるゼロウェイト及び1ウェイトのライトオペレーションのタイミングを示す。なお、単一転送では信号#BL 612がネゲートされ、信号#BURST 613は無視される。

【0248】図70乃至図73のタイミングチャートはバーストモードによるバスサイクルを示す。バーストモードは4ワードのデータを短時間に入出力するために使用され、ゼロウェイトのバーストモードでは5クロックサイクルに4ワードのデータを転送することが可能である。バーストモードは信号#BL 612がアサートされたバスサイクルの第1転送サイクルで信号#BURST 613をアサートすることにより指定され、1つのアドレス出力に対して同一の16バイト境界内の4ワードのデータが転送される。

【0249】バーストモードでは信号#BS 614がアサー

トされたタイミングの2クロック後のCLK 600 の立ち上がり時から各クロック毎に信号#DC 605 がサンプリングされ、信号#DC 605 の4回目のアサートの時点でバスサイクルが終了する。図70はバーストモードによるゼロウェイトのリードオペレーションのタイミングを示し、図71は第1サイクルが3ウェイト、第2～4サイクルが全て1ウェイトである場合のリードオペレーションのタイミングを示す。図72はバーストモードによるゼロウェイトのライトオペレーションのタイミングを示し、図73は第1サイクルが3ウェイト、第2～4サイクルが全て1ウェイトである場合のライトオペレーションのタイミングを示す。

【0250】(6)「ストリング操作命令、ビットマップ操作命令に関連するデータ演算部の構成」

図74及び図75は本発明のデータ処理装置100のストリング操作命令及びビットマップ操作命令の実行に必要な要部の構成を示すブロック図である。本発明のデータ処理装置100では、命令の実行制御をマイクロROM 320に格納されたマイクロ命令をマイクロシーケンサ321が順次読み出すことによって行なっており、以下に説明する各ブロックもマイクロ命令の制御によってその機能を果たす。まず、構成について説明する。

【0251】(6.1)「データ演算部の構成」

参照符号10, 11, 12, 13, 14, 15はいずれも32ビットバスであり、それぞれS1バス, S2バス, S3バス, S4バス, D1バス, D3バスと称される。参照符号16, 17は外部のメモリ装置、ステップ主記憶装置103をアクセスする際に、アクセスすべきアドレスをセットするための32ビットのメモリアドレスレジスタである。以下、参照符号16をAA1レジスタと称し、参照符号17をAA2レジスタと称す。AA1レジスタ16, AA2レジスタ17へのアドレスの設定及び読み出しはそれぞれS1バス10, S2バス11を介して行なわれる。また、AA1レジスタ16, AA2レジスタ17にはそれぞれカウント値指定レジスタ18, 19が備えられており、“1”, “2”, “4”, “8”のインクリメント／デクリメントが可能である。AA1レジスタ16及びAA2レジスタ17が示すメモリアドレス値はAAバス160からオペランドアドレスレジスタ299を経由してOAバス172へ出力され、バスインターフェース部157に入力される。

【0252】参照符号20は、30個の32ビットレジスタで構成されるレジスタファイルである。R0レジスタからR29レジスタで構成されている。各レジスタは、D1バス14及びD3バス15からの入力経路とS1バス10, S2バス11, S3バス12及びS4バス13への出力経路とを有する。第1ALUレジスタA 21と第1ALUレジスタB 22とは、「32ビット＋拡張ビット1ビット」の33ビット幅のレジスタであり、共にS1バス10及びS2バス11からの入力経路と第1ALU 23へ出力する経路とを有し、入力データをサイズ、符号に応じて33ビットデータに符号拡張または0拡張する。

【0253】第1ALU 23は、第1ALUレジスタA 21と第1ALUレジスタB 22とから送られてくるデータ間で加減算及び論理演算等を実行し、結果をM0レジスタ24へ出力する。この際、演算結果に応じた各フラグ情報を同時に割り出し検出フラグ40として出力する。フラグ情報はマイクロシーケンサ321に入力され、マイクロシーケンサは入力されたフラグ情報に従って、マイクロプログラムの実行シーケンスを制御する。また、第1ALU 23はストリング命令を効率よく処理するために8ビット単位でデータの一致、不一致を検出し、結果をM0レジスタ24へ出力する機能も合わせ持っている。第1ALU 23の機能の詳細は後述する。

【0254】第2ALUレジスタA 25と第2ALUレジスタB 26とは「32ビット＋拡張ビット1ビット」の33ビット幅のレジスタであり、第2ALUレジスタA 25はS1バス10, S2バス11, S3バス12から、第2ALUレジスタB 26はS2バス11, S4バス13からの入力経路を有し、またいずれも第2ALU 27への出力経路を有し、入力データをサイズ、符号に応じて33ビットデータに符号拡張または0拡張する。

【0255】第2ALU 27は、第2ALUレジスタA 25と第2ALUレジスタB 26とから送られてくるデータ間で加減算及び論理演算等を実行し、結果をS0レジスタ28へ出力する。この際、演算結果に応じた各フラグ情報を割り出し検出フラグ41として出力する。フラグ情報はマイクロシーケンサ321に入力され、マイクロシーケンサは入力されたフラグ情報に従ってマイクロプログラムの実行シーケンスを制御する。また、第2ALU 27はストリング命令を効率よく処理するために8ビット単位でデータの一致、不一致を検出し、結果をS0レジスタ(28)へ出力する機能も合わせ持っている。第1ALU 23と第2ALU 27とはマイクロプログラムがそれぞれ独立に制御出来るようになっているため同時に並列動作を行なう事が可能であり、1サイクルで64ビット(8バイト)のデータを処理することができる。

【0256】参照符号29はバレルシフタであり、S1バス10及びS2バス11から入力されたデータを連結して最大64ビット幅でのシフト演算を行なう。シフト結果はD1バス14及びD3バス15へ出力される。参照符号30, 31は、デクリメントであり、それぞれDECR1, DECR2と称す。DECR1, DECR2はS1バス10, S2バス11からの入力経路とD1バス14, D3バス15への出力経路とを有しており、カウント値指定レジスタ32, 33に格納された値に従って一度に“1”, “2”, “3”, “4”, “8”, “32”, “64”だけデクリメントすることが可能である。また、DECR1及びDECR2に格納された値が指定された値未満になった場合にその旨をフラグ信号34, 35で通知する機能を備えている。フラグ情報はマイクロシーケンサ321に入力され、マイクロシーケンサは入力されたフラグ情報に従って、マイクロプログラムの実行シーケンスを制御する。

る。

【0257】参照符号36、37は、メモリから読み込んだデータを一時的に格納するためのメモリデータレジスタ0とメモリデータレジスタ1であり、以下、DDRO、DDR1と称す。DDRO、DDR1はそれぞれが64ビット（8バイト）分の容量を有している。参照符号38はメモリへ書き込むデータを一時的に格納するためのレジスタであり、以下DDWと称する。このDDW 3864ビット（8バイト）分の容量を有している。参照符号39はストアバッファデータ部であり、SBDATAと称す。メモリに対するストアデータは、このSBDATA 39を介してBDバス163へ出力され、バスインターフェイス部157へ入力される。SBDATA 39は128ビット（16バイト）分の容量を有している。

【0258】参照符号290はデータキャッシュであり、4KBの容量を有している。メモリ上のデータをフェッチする際、OAバス172上のメモリアドレスがキャッシュヒットすると、データキャッシュ290はアクセスされたメモリデータをDDバス164へ出力し、DDRO 36またはDDR1 37がこのデータを格納する。データキャッシュ290の出力、DDバス164及びDDRO、DDR1は64ビット（8バイト）幅であるため、1サイクルで8バイトのデータをフェッチすることが可能である。

【0259】バスインターフェイス部157は、OAバス172から入力されるメモリアドレスでメモリをアクセスする。メモリとのデータアクセスは32ビット幅（4バイト）のバスD00:31 615で行なわれ、アドレスはバスA00:31 616へ出力される。演算部からメモリに対して16バイトブロック転送の要求を行なう場合、マイクロプログラムの指示により信号#BL 612をアサートする。メモリはこの要求に応じると信号#BURST 613をアサートしてバーストモードでの転送が可能であることをバスインターフェイス部157に通知する。

【0260】(6.2)「第1ALU及び第2ALUの機能」
前述したように、本発明のデータ処理装置は命令の実行制御をマイクロプログラム制御方式で行なっている。図76に第1ALU 23を制御するために設けられたマイクロプログラムフィールドと、そこで指定が可能なマイクロオペレーションの内容とを示す。また、図77に第2ALU 27を制御するために設けられたマイクロプログラムフィールドと、そこで指定が可能なマイクロオペレーションの内容とを示す。

【0261】maaiフィールドとmabiフィールドとは、それぞれ第1ALUレジスタA 21と第1ALUレジスタB 22との入力制御を指定する。maopフィールドは、第1ALU 23で実行される演算を定義する。算術演算、論理演算に加え、ストリング命令を効率良く処理するための一致／不一致演算機能を有している。一致／不一致演算機能の詳細は後述する。masuフィールドは、第1ALU 23の符号拡張／ゼロ拡張を指定する。第1ALU 23は、第1ALUレジスタA 21、または第1ALUレジスタB 22にオペランドを

取り込む際に符号拡張／ゼロ拡張を行なう。

【0262】maazフィールドは、第1ALU 23がS1バス10から第1ALUレジスタA 21にオペランドを取り込む際に符号／ゼロ拡張を行なうサイズを指定する。mabzフィールドは、第1ALU 23がS2バス11から第1ALUレジスタB 22にオペランドを取り込む際に符号／ゼロ拡張を行なうサイズを指定する。mafzフィールドは、第1ALU 23が演算結果に応じた各種フラグを出力する際のベースとなるサイズを指定する。maotフィールドは、第1ALU 23で実行された演算結果をD1バス14、D3バス15の何れに出力するかを指定する。saaiフィールドとsabiフィールドとは、それぞれ第2ALUレジスタA 25と第2ALUレジスタB 26との入力制御を指定する。

【0263】saopフィールドは、第2ALU 27で実行される演算を定義する。算術演算、論理演算に加え、ストリング命令を効率良く処理するための一致／不一致演算機能を有している。一致／不一致演算機能の詳細は後述する。sasufieldは、第2ALU 27の符号拡張／ゼロ拡張を指定する。第2ALU 27は、第2ALUレジスタA 25、または第2ALUレジスタB 26にオペランドを取り込む際に符号拡張／ゼロ拡張を行なう。

【0264】saazフィールドは、第2ALU 27がS1バス10、S2バス11、S3バス12から第2ALUレジスタA 25にオペランドを取り込む際に符号／ゼロ拡張を行なうサイズを指定する。sabzフィールドは、第2ALU 27がS2バス11、S4バス13から第2ALUレジスタB 26にオペランドを取り込む際に符号／ゼロ拡張を行なうサイズを指定する。saotフィールドは、第2ALU 27で実行された演算結果をD3バス15へ出力するか否かを指定する。

【0265】(6.3)「第1ALU、第2ALUにおける一致／不一致検出演算」

図78の模式図に第1ALU 23、第2ALU 27を使用した一致／不一致検出演算の手順を示す。メモリ上にストリングデータ列が存在しているものとする。このデータ列の任意の64ビット（8バイト）に着目した場合、各バイトに格納されているデータを図78に示されているように、“A”、“B”、“C”・・・“H”で表現する。一致／不一致検出演算とは、このストリングデータ列の中に所定のストリングが存在すること、あるいは存在しないことを調べるために用意された演算である。以下に演算の内容を説明する。

【0266】いまた例えば、所定のストリングを仮に1バイトサイズの“G”とする。一致検出演算を行なうか、あるいは不一致検出演算を行なうかの条件を「割り出し条件」と称し、いま割り出し条件として「一致」を指定するものとする。まず所定の1バイトストリング“G”をレジスタファイル20のR3に、図78に示されているように、4バイトサイズに展開しておく。次に、メモリから読み込んだ8バイトのストリングデータ列の内のアドレスの上位側4バイトを第1ALU 23に、下位側4バイ

トを第2 ALU 27にそれぞれ入力する。第1 ALU 23, 第2 ALU 27の演算は、図76に示されているmaopフィールドでeopf__eを、図77に示されているsaopフィールドでeopf__eをそれぞれ指定することにより順方向の一致演算が行なわれる。即ち、与えられたストリングデータ列の下位側のバイトから順に着目し、R3の値と一致するストリングが検出された場合にそのバイト位置を示す出力データをM0レジスタ24及びS0レジスタ28へ出力する。

【0267】図78に示した例の場合、第1 ALU 23に入力されたストリング列” EFGH ”の中に所定のストリング” G ”が検出されるので、第1 ALU 23は” G ”が検出された位置が8バイトデータの第6バイト目である事を示す32ビット出力” h00000006 ”をM0レジスタ24へ出力する。同時に所定のストリングが検出された事を示すフラグ情報を割り出し検出フラグ40として出力する。

【0268】一方、第2 ALU 27に入力されたストリング列” ABCD ”の中には所定のストリング” G ”が検出されないで、その旨を示す32ビット出力” hffffff ”をS0レジスタ28へ出力する。同時に、所定のストリングが検出されなかった旨を示すフラグ情報を41として出力する。

【0269】(7) 「SMOV命令処理の説明」
ストリング操作命令の一例として、本発明のデータ処理装置100が備えるSMOV命令を取り上げてその処理の手順を説明する。まず、SMOV命令の処理の概要について説明する。図79はSMOV命令の処理の手順を示す模式図である。

【0270】SMOV命令は、メモリのアドレスmから始まるp個のストリングデータ列をメモリのアドレスnから始まる領域へ順へ転送する処理を行なう。ストリング1個あたりのサイズは8ビット(1バイト)、16ビット(2バイト)、32ビット(4バイト)のいずれかを指定できる。但し、転送を行なっていく際、そのストリングデータが割り出し条件を満足するかどうかのチェックが行なわれ、割り出しが検出された場合はそのストリングを転送した時点で命令処理が終了する。いずれのストリングでも割り出しが検出されなければp個のデータを全て転送して命令処理を終了する。

【0271】図80はSMOV命令のオペランドを示す模式図である。SMOV命令のオペランドは、レジスタファイル20中のR0レジスタ、R1レジスタ、R2レジスタ、R3レジスタに格納される。R0レジスタにはソース側ストリングの先頭アドレスが、R1レジスタにはディスティネーション側の先頭アドレスが、R2レジスタにはストリングの長さを表すエレメント数が、R3レジスタには割り出し条件の比較値がそれぞれセットされる。

【0272】図81(a), (b)及び(c)はSMOV命令実行時にD1バス14からR3レジスタに比較値をセットする際のR3レジスタの動作を説明する模式図である。図81(a)に示すように、比較値のサイズが8ビットである場合、D1バス

14のデータD1(24:31)がR3レジスタのR3(0:7), R3(8:15), R3(16:23), R3(24:31)にそれぞれセットされる。また、図81(b)に示すように、比較値のサイズが16ビットの場合、D1バス14のデータD0(16:31)がR3レジスタのR3(0:15), R3(16:31)にそれぞれセットされる。更に、図81(c)に示すように、比較値のサイズが32ビットの場合、D0バス14のデータD0(0:31)がR3レジスタのR3(0:31)にセットされる。

【0273】以下のSMOV命令の処理の例では、操作の対象となるソースデータとディスティネーションデータとは全て32ビット(ワード)の整置境界上にあるデータに限定する。操作の対象となるデータが32ビット(ワード)の整置境界をまたぐ場合はバスインターフェース部157のメモリに対するアクセス回数が増えるため、読み出したデータと書き込むデータとに対してデータのシフト動作と連結動作を行なってデータを整置する必要があるが、その処理方法については、本願出願人が先に願願している特開昭64-88837号公報に詳しく開示されている。

【0274】(7.1) 「SMOV命令処理のアルゴリズム」
次に、本発明のデータ処理装置100により、1個のエレメントが8ビット(1バイト)で構成される224ビット(28バイト)サイズのストリングデータをSMOV命令でアドレスの増加方向に処理する動作を、図82及び図83のフローチャートを参照して説明する。本発明のデータ処理装置では、ストリング列の長さが一定値以上である場合、ストリングのエレメントサイズとは無関係に16バイトずつ処理する。ソースデータのフェッチと割り出しのチェックとは8バイトずつ2回に分けて行なわれ、ディスティネーションへのストアは16バイト単位のブロックストア動作で行なわれる。

【0275】いまたとえば、操作の対象となるストリングデータには割り出し条件を満足するようなエレメントが含まれていないものとする。また、ソースデータのアドレスを” m ”とし、ディスティネーションデータのアドレスを” n ”とし、R0レジスタとR1レジスタとにそれぞれセットされているものとする。なお、前述したようにm, nはそれぞれ32ビット(4バイト)の整置境界上にあるとし、更にnについては、nの次の32ビット境界が64ビット(8バイト)の整置境界上にあるようなアドレスであると仮定する。また、ストリングの長さを表すエレメント数” 28 ”がR2レジスタに、割り出し条件の比較値がR3レジスタにそれぞれセットされているものとする。更に、割り出し条件として「一致」が指定されているものとする。

【0276】第1ステップでは、R0レジスタの内容がAA1レジスタ16にセットされ、更にR2レジスタの内容がデクリメンタDECR1 30にセットされる。第2ステップでは、R1レジスタの内容がAA1レジスタ17にセットされ、R3レジスタに格納された割り出し比較値が図81に示され

ているように32ビットサイズに展開されて再びR3レジスタに格納される。

【0277】更に、ディスティネーションの次の8バイト境界までに相当するサイズ分だけデータがメモリからフェッチされてDDR036に格納される。ディスティネーションの先頭アドレスnは4バイト整置境界にあり、次の4バイト境界が8バイト整置境界にあたるので、メモリから読み出されるデータサイズは4バイトとなる。4バイトのデータが読み出されたため、DECR130とAA116との値を更新するためにカウント値指定レジスタ32及び18に“4”がセットされ、DECR1の値は“4”だけデクリメントされて“24”に、AA1の値は“4”だけインクリメントされて“m+4”となる。

【0278】第3ステップでは、第1ALU23、第2ALU27を使用して、前のステップでDDR036に格納された値の一致検出演算が行なわれる。但し、DDR036に格納されているデータは4バイトであるので、第2ALU27にはデータの inputs は行なわれない。一方、第1ALU23に inputs された4バイトのデータの中には割り出し条件を満たすエレメントは含まれていないので、第1ALU27はその旨をM0レジスタ24及び割り出し検出フラグ40として出力する。

【0279】第4ステップでは、DDR036の値がS1バス10、第1ALU23及びD1バス14を経由してDDW38へ転送され、後にディスティネーション側のメモリへ書き込むための準備が行なわれる。前のステップの割り出し検出フラグ40が“1”であるか、あるいはDECR130の値が8バイト未満であることを示すフラグ34の値が“1”であれば、マイクロプログラムによりシーケンスが制御され、命令の終了処理を行なうマイクロプログラムの実行アドレスに分岐する。ここで、DECR130の値は“20”であり、また前のステップで割り出しも検出されなかったため、終了処理への分岐は発生しない。

【0280】第5ステップでは、ソースデータが8バイト分メモリからフェッチされてDDR036に格納される。8バイトのデータを読み出したことによりDECR130とAA116との値を更新するため、カウント値指定レジスタ32及び18に“8”がセットされ、DECR1の値は“8”だけデクリメントされて“16”に、AA1の値は“8”だけインクリメントされて“(m+4)+8”となる。

【0281】第6ステップでは、第4ステップでストアデータレジスタ(割り出し検出フラグ40)に準備しておいたデータがAA2のアドレスに従ってメモリに書き込まれる。4バイトのデータが書き込まれたので、カウント値指定レジスタ(19)には“4”が指定され、AA2レジスタの値は4だけインクリメントされて“n+4”となる。更に、第1ALU23、第2ALU27を使用して、前のステップでDDR036に格納した値の一致検出演算が行なわれる。DDR036に格納されているデータは8バイトであるので、第1ALU23がメモリアドレスの上位側4バイト

の、第2ALU27がメモリアドレスの下位側4バイトの一致検出をそれぞれ受け持つ。検出対象の8バイトのデータの中には割り出し条件を満たすエレメントは含まれていないので、第1ALU23、第2ALU27はその旨をM0レジスタ24、S0レジスタ28及び割り出し検出フラグ40、41として出力する。

【0282】第7ステップでは、DDR036に格納された8バイトのデータが4バイトずつS1バス10→第1ALU23→D1バス14、及びS2バス11→第2ALU27→D3バス15の経路でDDW38へ転送され、後にディスティネーション側のメモリへ書き込むための準備が行なわれる。前のステップで割り出しが検出されたか、あるいはDECR130の値が8バイト未満であることを示すフラグ34の値が“1”であれば、マイクロプログラムによりシーケンスが制御され、命令の終了処理を行なうマイクロプログラムの実行アドレスに分岐する。ここで、DECR130の値は“16”であり、前のステップで割り出しも検出されなかったため、終了処理への分岐は発生しない。

【0283】第8ステップでは、ソースデータが8バイト分メモリからフェッチされてDDR137に格納される。8バイトのデータが読み出されたことによりDECR130とAA116との値を更新するために、カウント値指定レジスタ32及び18に“8”がセットされ、デクリメンタの値は“8”だけデクリメントされて“8”に、AA1の値は“8”だけインクリメントされて“(m+4+8)+8”となる。

【0284】第9ステップでは、第1ALU23、第2ALU27を使用して、前のステップでDDR137に格納された値の一致検出演算が行なわれる。DDR137に格納されているデータは8バイトであるので、第1ALU23がメモリアドレスの上位側4バイトの、第2ALU27がメモリアドレスの下位側4バイトの一致検出をそれぞれ受け持つ。検出対象の8バイトのデータの中には割り出し条件を満たすエレメントは含まれていないので、第1ALU23、第2ALU27はその旨をM0レジスタ24、S0レジスタ28及び割り出し検出フラグ40、41として出力する。

【0285】第10ステップにおいて、前のステップで割り出しが検出されたか、あるいはDECR130の値が8バイト未満であることを示すフラグ34の値が“1”であればマイクロプログラムによりシーケンスが制御されて命令の終了処理を行なうマイクロプログラムの実行アドレスに分岐する。ここで、DECR130の値は“8”であり、前のステップで割り出しも検出されなかったため終了処理への分岐は発生しない。

【0286】第11ステップでは、ソースデータが8バイト分メモリから読み出されてDDR036に格納される。8バイトのデータが読み出されたことによりDECR130とAA116との値を更新するために、カウント値指定レジスタ32及び18に“8”がセットされ、デクリメンタの値は“8”だけデクリメントされて“0”に、AA1レジス

10

20

30

40

50

タの値はは”8”だけインクリメントされて”(m+4+8+8+8)”となる。第12ステップでは、第7ステップでDDW 38に準備しておいた8バイトのデータがメモリにストアされる。この際、ブロックストアを行なうことをメモリに通知する。更に、AA2レジスタのカウント値指定レジスタ19に”8”が指定され、AA2レジスタの値は8だけインクリメントされて”(n+4)+8”となる。

【0287】更に、第1ALU 23、第2ALU 27を使用して、前のステップでDDR0 36に格納された値の一致検出が行なわれる。DDR0 36に格納されているデータは8バイトであるので、第1ALU 23がメモリアドレスの上位側4バイトの、第2ALU 27がメモリアドレスの下位側4バイトの一致検出をそれぞれ受け持つ。検出対象の8バイトのデータの中には割り出し条件を満たすエレメントは含まれていないので、第1ALU 23、第2ALU 27はその旨の出力をM0レジスタ24、S0レジスタ28及び割り出し検出フラグ40、41として出力する。

【0288】第13ステップでは、DDR1 37に格納された8バイトのデータが4バイトずつS1バス10→第1ALU 23→D1バス14、及びS2バス11→第2ALU 27→D3バス15の経路でDDW 38へ転送されてそのままメモリにストアされる。前のステップでブロックストアが指定されているため、この書き込みはブロックストアの引き続くデータと見なされる。AA2レジスタのカウント値指定レジスタ19に”8”が指定され、AA2レジスタの値は8だけインクリメントされて”(n+4+8)+8”となる。以上の処理を行なった後、第7ステップへ分岐する。

【0289】(7.2)「SMOV命令処理における外部バスサイクル」
上述のようなアルゴリズムを用いたSMOV命令の処理では、転送を行なうべきストリングの長さが十分に大きく、かつ割り出しの検出が無い場合には、図82及び図83のフローチャートに従って、第7ステップ～第13ステップの処理を反復しつつストリングデータ列を転送していくことになる。

【0290】図84乃至図87のタイミングチャートに、上述の場合の本発明のデータ処理装置100におけるバーストライト時の外部バスサイクルを示す。図84はメモリのウェイト数が0-0である場合に、図85はウェイト数が1-0である場合に、図86はウェイト数が2-0である場合に、図87はウェイト数が3-0である場合にそれぞれ対応している。またソースのストリングデータ列は本発明のデータ処理装置100が内蔵するデータキャッシュに予め格納されているため、ソースデータのフェッチに関するメモリ装置へのアクセスは発生しないものとする。

【0291】図84乃至図87において、CLKは本発明のデータ処理装置100を駆動するためのクロック信号を、D0:31はメモリに書き込まれるデータを、#BLはブロック

転送要求信号を、#BURSTはメモリから出力されるバースト転送アクノレッジ信号を、#DCはメモリから出力されるデータコンプリート信号をそれぞれ示している。本発明のデータ処理装置100において、マイクロプログラムがブロックストアを指定できるのは、直前に指定したブロックストアの最後のデータの書き込みが完了しており、信号#DCをメモリから受け取った次のクロックサイクルにおいてである。従って、メモリのウェイト数が3以上になった場合には、次のブロックストアをマイクロプログラムが指示する第17ステップの処理へ進むまでに図86及び図87に示されているような空きサイクルが自動的に挿入される。以上のように、SMOV命令の実行に際して、128ビット(16バイト)のビットマップデータ列の転送が計8ステップのマイクロ命令による制御を反復することによって実現する本発明のデータ処理装置100のデータ処理効率は各ウェイト数に対して図88の一覧表に示されているようになる。

【0292】(8)「BVMAP命令処理の説明」

ビットマップ操作命令の一例として、本発明のデータ処理装置100が備えるBVMAP命令を取り上げ、処理の手順を説明する。まず、BVMAP命令の処理の概要について説明する。図89はBVMAP命令の処理手順を示す模式図である。

【0293】BVMAP命令は、メモリのアドレスjからkだけのオフセットを有するアドレスから始まるwビット分のソースビットマップデータ列と、メモリのアドレスmからnだけのオフセットを持ったアドレスから始まるwビット分のディスティネーションビットマップデータとの間で、1ビット単位で所定の演算を行なった後にディスティネーション側のビットマップデータ領域へ順に転送する処理を行なう。但し、wとして、0以下の値が指定された場合には何等の処理を行わずに命令を終了する。

【0294】図90はBVMAP命令のオペランドを示す模式図である。BVMAP命令のオペランドは、レジスタファイル20中のR0レジスタ、R1レジスタ、R2レジスタ、R3レジスタ、R4レジスタ、R5レジスタに格納される。R0レジスタにはソース側ビットフィールドのアドレスが、R1レジスタにはソース側ビットフィールドのビットオフセットが、R2レジスタには処理を行なうビットフィールドの長さ(=ビット数)が、R3レジスタにはディスティネーション側ビットフィールドのアドレスが、R4レジスタにはディスティネーション側ビットフィールドのビットオフセットが、R5レジスタには演算の種類を指定する情報がそれぞれセットされる。

【0295】(8.1)「BVMAP命令処理のアルゴリズム」
次に、本発明のデータ処理装置により、図91に示すような十分に長いビットマップデータ列をBVMAP命令でアドレスの増加方向に処理する動作を図92、図93及び図94のフローチャートを参照して説明する。本発明のデータ処

理装置では、ビットマップデータ列の長さが一定値以上である場合、16バイトづつ処理する。ソースデータのフェッチと処理終了のチェックとは8バイトづつ2回に分けて行なわれ、ディスティネーションへのストアは16バイトのブロックストア動作で行なわれる。

【0296】いまた例えば、ソースデータのアドレスを”j”、オフセットを”k”、ディスティネーションデータのアドレスを”m”、オフセットを”1”とし、R0レジスタ、R1レジスタ、R2レジスタ、R3レジスタにそれぞれセットされているものとする。なお、”m+1”で与えられるディスティネーション側のビットフィールドの先頭アドレスは64ビット（8バイト）の整置境界上にあるとする。更に、処理を行なうビットフィールドの長さ”width”がR2レジスタに、演算指定として”and”を指定するビットパターン情報がR5レジスタにそれぞれセットされているものとする。

【0297】第1ステップでは、R2レジスタに格納された”width”の値がゼロ以下の値であるか否かがチェックされる。更に、R1レジスタに格納されたソースアドレス”j”とオフセット”k”との正規化が行なわれる。ここで、アドレスの正規化というのは、ビットフィールドの最下位ビット位置をそのビットを含む含む64ビット（8バイト）アライメントのとれたアドレスと、このアドレスに対するオフセット値とで表す操作のことである。具体的には、アドレスの正規化は、「アドレス値+オフセット値/64の商」なる値の下位3ビットをゼロに変換することによって行なわれる。また、オフセットの正規化は、「オフセット/64」の剰余を求めることによって行なわれる。求められた正規化アドレスはAA2 17にセットされ、正規化オフセットをD_OFFSETとする。

【0298】第2ステップでは、R2に格納された”width”の値がDEC2 31にセットされる。DEC2の値は処理の進捗に従ってデクリメントされ、DEC2の値がディスティネーション側ビットマップデータの未処理のビット数を表すようにする。更に、R3レジスタに格納されたソースアドレス”m”とオフセット”1”との正規化が行なわれる。求められた正規化アドレスがAA1 16にセットされ、正規化オフセットをS_OFFSETとする。前のステップにおけるチェックで”width” ≤ 0と判定された場合にはこのまま命令の処理が終了する。

【0299】第3ステップでは、R2に格納された”width”の値がDEC1 30にセットされる。DEC1の値は処理の進捗に従ってデクリメントされ、DEC1の値がソース側ビットマップデータの未処理のビット数を表すようにする。更に、S_OFFSETとD_OFFSETとの差が算出され、その結果を△とする。第4ステップでは、AA1 16のアドレスに従ってソースデータの先頭部分を含むように64ビット（8バイト分。図91のn-4、n-3の部分）だけフェッチされてDDR0 36に格納する。このデータをsrc(n-4)とする。更に、AA1 16とDEC1 30とがフェ

ッチしたソースデータの分だけ更新される。即ち、AA1 16は”8”だけインクリメントされ、DEC1 30は64だけデクリメントされる。

【0300】第5ステップでは、src(n-4)がバレルシフタ29を使用して△だけ左シフトされる。この結果をsrc(n-4)とする。更に、ソースデータが8バイトだけメモリからフェッチされてDDR0 36に格納される。このデータをsrc(n-2)とする。更に、AA1 16とDEC1 30との値が64ビット（8バイト）分だけ更新される。第6ステップでは、src(n-2)がバレルシフタ29を使用して(64-△)だけ右シフトされる。この結果をsrc(n-2)とする。

【0301】第7ステップでは、src(n-4)とsrc(n-2)との論理和が求められる。このデータをsrc(n-4 | n-2)とする。第8ステップでは、src(n-2)がバレルシフタ29を使用して△だけ左シフトされる。この結果をsrc(n-2)とする。更に、ソースデータが8バイトだけメモリからフェッチされてDDR0 36に格納される。このデータをsrc(n)とする。更に、AA1 16とDEC1 30との値が64ビット（8バイト）分だけ更新される。

【0302】第9ステップでは、src(n)がバレルシフタ29を使用して(64-△)だけ右シフトされる。この結果をsrc(n)とする。更に、ディスティネーションデータの先頭部分が8バイトだけメモリからフェッチされ（図91のn-4、n-3の部分）、DDR0 36に格納される。このデータをdest(n-4)とする。第10ステップではsrc(n-4 | n-2)とdest(n-4)との間でR5レジスタに指定された演算が行なわれ、その結果がDDW 38に格納されてそのままディスティネーション側に8バイトストアする。更に、AA2 16とDEC2 31との値が64ビット（8バイト）分だけ更新される。即ち、AA2 16が”8”だけインクリメントされ、DEC2 31の値が”64”だけデクリメントされる。

【0303】第11ステップでは、src(n-2)とsrc(n)との論理和が求められ、このデータをsrc(n-2 | n)とする。第12ステップではsrc(n)がバレルシフタ29を使用して△だけ左シフトされる。この結果をsrc(n)とする。更に、ソースデータが8バイトだけメモリからフェッチされてDDR 36に格納される。このデータをsrc(n+2)とする。更に、AA1 16とDEC1 30との値が64ビット（8バイト）分だけ更新される。

【0304】第13ステップでは、src(n+2)がバレルシフタ29を使用して(64-△)だけ右シフトされる。この結果をsrc(n+2)とする。更に、ディスティネーションデータが8バイトだけメモリからフェッチされてDDR0 36に格納される。このデータをdest(n-2)とする。第14ステップでは、src(n-2 | n-1)とdest(n-2)との間でR5レジスタに指定された演算が行なわれ、その結果がDDW 38に格納される。

【0305】第15ステップでは、src(n)とsrc(n+2)との論理和が求められる。このデータをsrc(n | n+2)とする。更に、ソースデータが8バイトだけメモリか

10

20

30

40

50

らフェッチされてDDRO 36 に書き込まれる。このデータをsrc(n+4)とする。更に、AA1 16とDECR1 30との値が64ビット(8バイト)分だけ更新される。第16ステップでは、src(n+2)がバレルシフタ29を使用して△だけ左シフトされる。この結果をsrc■(n+2)とする。更に、ディステーションデータが8バイトだけメモリからフェッチされてDDR1 37 に書き込まれる。このデータをdest(n)とする。

【0306】第17ステップでは、src(n+4)がバレルシフタ29を用いて(64-△)だけ右シフトされる。この結果をsrc■■(n+4)とする。更に、第14ステップでDDW 38に格納しておいたデータがストアバッファに8バイトストアされ、同時にメモリへの16バイトブロック転送が指定される。第18ステップでは、src(n | n+2)とdest(n)との間でR5レジスタに指定された演算が行われ、その結果がDDW 38に格納されてそのままストアバッファに8バイトストアされる。直前のステップで16バイトブロック転送を指定しているので、本ステップのストアも引き続きブロック書き込みと見なされる。

【0307】第19ステップでは、src■(n+2)とsrc■■(n+4)との論理和が求められる。このデータをsrc(n+2 | n+4)とする。更に、n=n+4とした上で、第12ステップに分岐する。

【0308】(8.2)「BVMAP命令処理における外部バスサイクル」

上述のようなアルゴリズムを用いたBVMAP命令の処理では、処理を行なうべきビットマップフィールドの長さが十分に大きい場合には、図92、図93及び図94のフローチャートに従って、第12ステップ～第19ステップの処理を反復しつつビットマップデータ列を処理していくことになる。図95乃至図98のタイミングチャートに、上述の場合の本発明のデータ処理装置100における外部バスサイクルを示す。

【0309】図95はメモリのウェイト数が0-0である場合に、図96はウェイト数が0-1である場合に、図97はウェイト数が0-2である場合に、図98はウェイト数が0-3である場合にそれぞれ対応している。また処理の対象となるビットマップデータ列は本発明のデータ処理装置100が内蔵するデータキャッシュ290に予め格納されているため、データフェッチに関するメモリ装置へのアクセスは発生しないものとする。

【0310】本発明のデータ処理装置100において、マイクロプログラムがブロックストアを指示できるのは、直前に指定したブロックストアの最後のデータの書き込みが完了しており、信号#DCをメモリから受け取った次のクロックサイクルにおいてである。従って、メモリのウェイト数が3以上になった場合には、次のブロックストアをマイクロプログラムが指示する第17ステップの処理へ進むまでに図98のような空きサイクルが自動的に挿入される。

【0311】以上のように、BVMAP命令の実行に際して、128ビット(16バイト)のビットマップデータ列の転送を計8ステップのマイクロ命令による制御を反復することによって実現する本発明のデータ処理装置100のデータ処理効率は各ウェイト数に対して図99の一覧表に示されているようになる。

【0312】(9)「本発明の他の実施例」

(9.1)「データキャッシュを内蔵しないデータ処理装置への適用」

10 以上に説明した本発明の実施例では、データ処理装置がデータキャッシュを内蔵している事を前提とした。しかし装置のコストを考慮した場合、データキャッシュを内蔵しない処理装置も想定される。また、データキャッシュを内蔵した本発明のデータ処理装置においても、その動作を無効に設定できる機能を備えており、データキャッシュを無効にした場合はそれが内蔵されていないデータ処理装置と同じ動作をすることになる。

【0313】そのような場合でも、外部のメモリからデータをフェッチする場合に、メモリリード時のバースト転送機能を使用して予め連続した16バイト分のデータをフェッチし、DDRO 36及びDDR1 37の二つのメモリデータレジスタに一時的に保持しておく事により、図82乃至図83及び図92乃至図94にそれぞれ示した処理アルゴリズムを変更すること無しにそのまま命令処理を行なうことが可能である。即ち、図82乃至図83及び図92乃至図94にそれぞれ示した繰り返し処理を行なう部分では、8バイトサイズのメモリデータフェッチを2回行なっているが、1回目のフェッチ時にバースト転送を要求するようにする。バスインターフェース部157はこの要求を受けてフェッチすべき8バイトのデータと次の8バイトのデータとを連続してバースト転送を使って取り込んでおく。この場合、2回目のフェッチは既に取り込んだデータの後半を使えばよいので、外部メモリへのバスアクセスは発生しない。

【0314】従って、外部メモリのウェイト数が大きい場合でも、4バイト単位あるいは8バイト単位でデータフェッチ→演算処理→データストアを行なう場合よりも高性能な処理を行なう事が出来る。

【0315】(9.2)「整置されていないデータ列に対する処理」

以上に説明した本発明の実施例では、その処理方法を端的に説明するために、ストリングデータ列あるいはビットマップデータ列が4バイト境界、8バイト境界等に整置されていることを前提とした。但し、ストリング操作命令、ビットマップ操作命令のいずれにおいても処理すべきデータ列を整置境界に置くような制約は設けていないため、当然アライメントがとれていないデータを処理する必要が生じる。

【0316】アライメントがとれていないデータを処理する場合でも、最初の16バイト整置境界までのデータだ

けを予め別途処理する事により、図82乃至図83及び図92乃至図94にそれぞれ示されているような繰り返し処理を行なう部分のアルゴリズムを変更すること無しに処理を実行する事が可能である。また、その場合の処理性能は図90及び図99に示したものと等しい。整置されていないデータを処理する場合の処理の詳細は本願出願人が先に出願している特開昭64-88837号公報に詳しく開示されている。

【0317】

【発明の効果】以上に詳述したように本発明のデータ処理装置によれば、バイト列またはビット列を外部のメモリのソース領域からデスティネーション領域へ転送するストリング転送命令またはビットマップ操作命令を命令デコーダがデコードした場合に、実行制御部の制御により、命令実行部がバイト列またはビット列をたとえば64ビット単位のデータ群に分解してそれぞれのデータ群をデータレジスタとアドレスレジスタとを用いてバスインターフェイス部へ入出力し、バスインターフェイス部がバースト転送によりデスティネーション領域をアクセスして二つのデータ群からなる128ビットのデータを一括して転送するため、連続したメモリ領域に格納されているバイト列またはビット列を高速で入出力することが可能になる。このため、ストリング転送命令またはビットマップ操作命令を高効率で実行出来る。

【0318】また本発明のデータ処理装置によれば、たとえば32ビットのパターンデータ複数を外部のメモリのデスティネーション領域に書き込むパターン充填命令を命令デコーダがデコードした場合に、実行制御部の制御により、命令実行部がデータレジスタとアドレスレジスタとを用いて二つのパターンデータを並列にバスインターフェイス部へ出力し、バスインターフェイス部がバースト転送により外部のメモリのデスティネーション領域に一度に4個のパターンデータを書き込むことが出来る。このため、パターンデータを外部メモリの連続した領域に高速に書き込むことが出来るので、パターン充填命令を高効率で実行出来る。

【0319】更に、本発明のデータ処理装置によれば、レジスタ群に保持されている内部状態を示すデータを前記外部メモリの任意のメモリ領域に連続して格納するプッシュ命令をデコードした場合に実行制御部の制御により、命令実行部がレジスタ群からデータをたとえば64ビット単位でデータレジスタとアドレスレジスタとを用いてバスインターフェイス部へ出力し、バスインターフェイス部がバースト転送によりデスティネーション領域をアクセスして転送し、またこのようにして外部メモリの連続する領域に格納されたデータをレジスタへ転送して保持させるポップ命令を命令デコーダがデコードした場合に、実行制御部の制御により、バスインターフェイス部が外部メモリのソース領域からデータをバースト転送によりたとえば64ビット単位でデータレジスタとアドレ

スレジスタとを用いてバスインターフェイス部に入力し、バスインターフェイス部がレジスタ群をアクセスして転送するため、プッシュ命令、ポップ命令を高速で実行することが可能になる。

【図面の簡単な説明】

【図1】本発明のデータ処理装置を使用したシステム構成を示すブロック図である。

【図2】本発明のデータ処理装置のメモリシステムの構成を示すブロック図である。

【図3】本発明のデータ処理装置のメモリシステムのアドレスの割付けを示す模式図である。

【図4】本発明のデータ処理装置における命令フォーマットを示す模式図である。

【図5】本発明のデータ処理装置における命令フォーマットを示す模式図である。

【図6】本発明のデータ処理装置における命令フォーマットを示す模式図である。

【図7】本発明のデータ処理装置における命令フォーマットを示す模式図である。

【図8】本発明のデータ処理装置のSMOV, SCMP, SSCH, SSTRの各命令のビットパターンと各種オプションを示す模式図である。

【図9】本発明のデータ処理装置のSMOV, SCMP, SSCH, SSTRの各命令のビットパターンと各種オプションを示す模式図である。

【図10】本発明のデータ処理装置のSMOV, SCMP, SSC H, SSTRの各命令のビットパターンと各種オプションを示す模式図である。

【図11】本発明のデータ処理装置のSMOV, SCMP, SSC H, SSTRの各命令のビットパターンと各種オプションを示す模式図である。

【図12】本発明のデータ処理装置のBVSCH, BVMAP, BVCPY, BVPATの各命令のビットパターンと各種オプションを示す模式図である。

【図13】本発明のデータ処理装置のBVSCH, BVMAP, BVCPY, BVPATの各命令のビットパターンと各種オプションを示す模式図である。

【図14】本発明のデータ処理装置のBVSCH, BVMAP, BVCPY, BVPATの各命令のビットパターンと各種オプションを示す模式図である。

【図15】本発明のデータ処理装置のBVSCH, BVMAP, BVCPY, BVPATの各命令のビットパターンと各種オプションを示す模式図である。

【図16】本発明のデータ処理装置の構成例を示すブロック図である。

【図17】本発明のデータ処理装置のバスインターフェイス部の構成例を示すブロック図である。

【図18】本発明のデータ処理装置の命令フェッチ部の構成例を示すブロック図である。

【図19】本発明のデータ処理装置における命令フォー

マットを示す模式図である。

【図 2 0】本発明のデータ処理装置の命令デコード部の構成例を示すブロック図である。

【図 2 1】本発明のデータ処理装置の PC 生成部の構成例を示すブロック図である。

【図 2 2】本発明のデータ処理装置のアドレス生成部の構成例を示すブロック図である。

【図 2 3】本発明のデータ処理装置のオペランドアクセス部の構成例を示すブロック図である。

【図 2 4】本発明のデータ処理装置の ROM 部の構成例を示すブロック図である。 10

【図 2 5】本発明のデータ処理装置の整数演算部の構成例を示すブロック図である。

【図 2 6】本発明のデータ処理装置の整数演算部の周辺の構成例を示すブロック図である。

【図 2 7】本発明のデータ処理装置の命令パイプラインの概念を示す模式図である。

【図 2 8】本発明のデータ処理装置においてスーパースケーラ処理を行なう場合のタイミングチャートである。

【図 2 9】本発明のデータ処理装置において並列デコードが可能な命令コードの組合せ条件を示す模式図である。 20

【図 3 0】本発明のデータ処理装置の ROM 部と整数演算部との並列実行機構に関する部分を示すブロック図である。

【図 3 1】本発明のデータ処理装置の VLIW 技法を用いたマイクロ命令により制御される整数演算部の構成例を示すブロック図である。

【図 3 2】本発明のデータ処理装置のレジスタリストを有する命令の例である LDM 命令のビットパターンを示す模式図である。 30

【図 3 3】本発明のデータ処理装置によりたとえば「LDM #SP+, (R4-R11)」命令を実行する際に、SP=H■(A+4) で A がダブルワード境界のアドレスである場合に転送対象となるオペランドを示す模式図である。

【図 3 4】本発明のデータ処理装置のレジスタリストから 2 つ連続した "1" の位置をエンコードするための構成を示すブロック図である。

【図 3 5】本発明のデータ処理装置において 2 オペランドを同時に並列転送可能な条件を示す表である。 40

【図 3 6】本発明のデータ処理装置コンテキストスイッチの際に使用されるレジスタ群であるコンテキストブロックを示すブロック図である。

【図 3 7】本発明のデータ処理装置の SMOV 命令を処理するマイクロプログラムの第 n 回目の 16 バイト処理のループの前半の 8 バイト処理の状態を示す模式図である。

【図 3 8】本発明のデータ処理装置による各ストリング命令の処理速度のピーク値の一覧表である。

【図 3 9】本発明のデータ処理装置による任意長ビットフィールド命令の処理速度のピーク値の一覧表を示す。 50

【図 4 0】本発明のデータ処理装置において先行ジャンプ処理を行なう命令と、先行ジャンプを行なうために使用するハードウェアとの対応を示す一覧表である。

【図 4 1】本発明のデータ処理装置のジャンプ命令の処理を行なうステージの関係を示すブロック図である。

【図 4 2】本発明のデータ処理装置において無条件分岐命令(BRA) に対して E ステージでジャンプを行なう場合のパイプライン中の命令の流れを示すタイミングチャートである。

【図 4 3】本発明のデータ処理装置において無条件分岐命令(BRA) に対して D ステージでジャンプを行なう場合のパイプライン中の命令の流れを示すタイミングチャートである。

【図 4 4】本発明のデータ処理装置の各ジャンプ命令に対する先行ジャンプ処理の有無と最小処理時間とを示す一覧表である。

【図 4 5】本発明のデータ処理装置において PC 相対アドレスへのジャンプに対してアドレスが計算される 4 種類の pcdisp フィールドと、絶対アドレスへの JMP, JSR 命令に対して切り出しと符号拡張処理とが行なわれる 2 種類の abs フィールドとを示す模式図である。

【図 4 6】本発明のデータ処理装置の Bcc 命令の実行履歴を記憶するための構成例を示すブロック図である。

【図 4 7】本発明のデータ処理装置の PC スタックの構成例を示すブロック図である。

【図 4 8】本発明のデータ処理装置において RTS 命令で先行リターンを行なった際のパイプライン中の命令の流れを示すタイミングチャートである。

【図 4 9】本発明のデータ処理装置において RTS 命令で先行リターンを行なった際のパイプライン中の命令の流れを示すタイミングチャートである。

【図 5 0】本発明のデータ処理装置の命令プリフェッチキューの構成例を示すブロック図である。

【図 5 1】本発明のデータ処理装置において BNE 命令に対して間違った分岐予測に従って先行分岐を行なった場合のパイプライン中の命令の流れを示すタイミングチャートである。

【図 5 2】本発明のデータ処理装置において BNE 命令に対して間違った分岐予測に従って先行分岐を行なった場合のパイプライン中の命令の流れを示すタイミングチャートである。

【図 5 3】本発明のデータ処理装置のオペランドアドレス生成機構の構成例を示すブロック図である。

【図 5 4】本発明のデータ処理装置のスコアボードレジスタの構成例を示すブロック図である。

【図 5 5】本発明のデータ処理装置において RAW データハザードによるパイプラインインターロックにより A ステージ 402 で命令がストールする場合パイプライン処理の流れを示すタイミングチャートである。

【図 5 6】本発明のデータ処理装置においてコンパイラ

がパイプラインストールを回避するために命令の順序を変更した場合のパイプライン処理の流れを示すタイミングチャートである。

【図57】本発明のデータ処理装置100においてスタックプッシュとスタックポップとに伴うRAWデータハザードを回避するために各パイプラインステージに備えられている作業用スタックポインタを示す模式図である。

【図58】本発明のデータ処理装置の先行命令のオペランドと後続命令のアドレッシングモードとの組み合わせと、各組み合わせに対するパイプラインストールの有無の関係を示す一覧表である。

【図59】本発明のデータ処理装置のプリフェッチしたオペランドを格納する2エントリのオペランドプリフェッチキュー291の構成例を示すブロック図である。

【図60】本発明のデータ処理装置においてキャッシュヒットの場合にストアアドレス、ストアデータ、PC値、ストア情報が格納されるストアバッファの構成例を示すブロック図である。

【図61】本発明のデータ処理装置においてメモリオペランド間のRAWデータハザードを避けるためパイプライン中で後続命令がストールする場合のパイプライン処理の流れをそれぞれ示すタイミングチャートである。

【図62】本発明のデータ処理装置においてコンパイラが命令の順序を入れ換えてこのストールを回避した場合のパイプライン処理の流れを示すタイミングチャートである。

【図63】本発明のデータ処理装置野メモリオペランド間のRAWデータハザードによるストール時間を示す一覧表である。

【図64】本発明のデータ処理装置において、無条件ジャンプ命令に引き続く命令（非ジャンプ側命令）の処理を行なわないことにより、非合理的なアドレスへのリードオペレーションを回避する命令列を示す模式図である。

【図65】本発明のデータ処理装置において、条件ジャンプを行なう命令(Bcc, ACB, SCB, TRAP)ではジャンプ、非ジャンプが確定するまでその後全ての命令のオペランドプリフェッチを外部のメモリからは行なわないことにより、非合理的なアドレスへのリードオペレーションを回避する命令列を示す模式図である。

【図66】本発明のデータ処理装置において、条件ジャンプ行なう命令(Bcc, ACB, SCB, TRAP)ではジャンプ、非ジャンプが確定するまでその後のジャンプ命令では絶対アドレス、PC相対以外のジャンプ先アドレス計算処理を行なわないことにより、非合理的なアドレスへのリードオペレーションを回避する命令列を示す模式図である。

【図67】本発明のデータ処理装置入出力信号を示す模式図である。

【図68】本発明のデータ処理装置の基本的なバスオペレーションの内の、単一転送によるゼロウエイト及び1ウエイトのリードオペレーションのタイミングを示すタ

イミングチャートである。

【図69】本発明のデータ処理装置の基本的なバスオペレーションの内の、単一転送によるゼロウエイト及び1ウエイトのライトオペレーションのタイミングを示すタイミングチャートである。

【図70】本発明のデータ処理装置の基本的なバスオペレーションの内の、バーストモードによるゼロウエイトのリードオペレーションのタイミングを示すタイミングチャートである。

【図71】本発明のデータ処理装置の基本的なバスオペレーションの内の、第1サイクルが3ウエイト、第2～4サイクルが全て1ウエイトである場合のリードオペレーションのタイミングを示すタイミングチャートである。

【図72】本発明のデータ処理装置の基本的なバスオペレーションの内の、バーストモードによるゼロウエイトのライトオペレーションのタイミングを示すタイミングチャートである。

【図73】本発明のデータ処理装置の基本的なバスオペレーションの内の、バーストモードによる第1サイクルが3ウエイト、第2～4サイクルが全て1ウエイトである場合のライトオペレーションのタイミングを示すタイミングチャートである。

【図74】本発明のデータ処理装置のストリング操作命令及びビットマップ操作命令の実行に必要な要部の構成を示すブロック図である。

【図75】本発明のデータ処理装置のストリング操作命令及びビットマップ操作命令の実行に必要な要部の構成を示すブロック図である。

【図76】本発明のデータ処理装置の第1ALUを制御するために設けられたマイクロプログラムフィールドと、そこで指定が可能なマイクロオペレーションの内容とを示す一覧表である。

【図77】本発明のデータ処理装置の第2ALUを制御するために設けられたマイクロプログラムフィールドと、そこで指定が可能なマイクロオペレーションの内容とを示す一覧表である。

【図78】本発明の第1ALU,第2ALUを使用した一致／不一致検出演算の手順を示す模式図である。

【図79】本発明のデータ処理装置によるSMOV命令の処理の手順を示す模式図である。

【図80】本発明のデータ処理装置のSMOV命令のオペランドを示す模式図である。

【図81】本発明のデータ処理装置によるSMOV命令実行時にD1バスからR3レジスタに比較値をセットする際のR3レジスタの動作を説明する模式図である。

【図82】本発明のデータ処理装置によるSMOV命令の実行手順を示すフローチャートである。

【図83】本発明のデータ処理装置によるSMOV命令の実行手順を示すフローチャートである。

【図 8 4】本発明のデータ処理装置による SMOV 命令の実行に際してメモリのウェイト数が 0-0 である場合のバーストライト時の外部バスサイクルを示すタイミングチャートである。

【図 8 5】本発明のデータ処理装置による SMOV 命令の実行に際してメモリのウェイト数が 1-0 である場合のバーストライト時の外部バスサイクルを示すタイミングチャートである。

【図 8 6】本発明のデータ処理装置による SMOV 命令の実行に際してメモリのウェイト数が 2-0 である場合のバーストライト時の外部バスサイクルを示すタイミングチャートである。

【図 8 7】本発明のデータ処理装置による SMOV 命令の実行に際してメモリのウェイト数が 3-0 である場合のバーストライト時の外部バスサイクルを示すタイミングチャートである。

【図 8 8】本発明のデータ処理装置による SMOV 命令の実行時のデータ処理効率の各ウェイト数に対する一覧表である。

【図 8 9】本発明のデータ処理装置による BVMAP 命令の 20 実行手順を示す模式図である。

【図 9 0】本発明のデータ処理装置の BVMAP 命令のオペランドを示す模式図である。

【図 9 1】本発明のデータ処理装置の BVMAP 命令による処理対象となる十分に長いビットマップデータ列の例を示す模式図である。

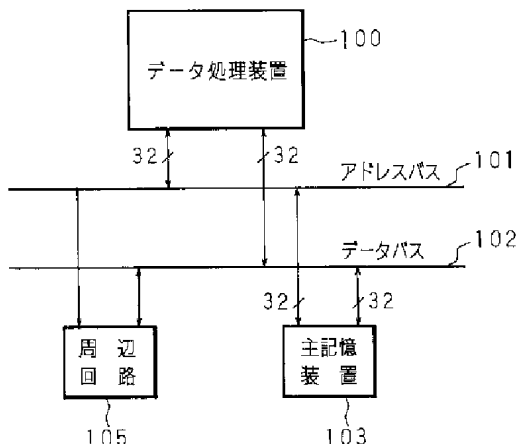
【図 9 2】本発明のデータ処理装置による BVMAP 命令の実行手順を示すフローチャートである。

【図 9 3】本発明のデータ処理装置による BVMAP 命令の実行手順を示すフローチャートである。

【図 9 4】本発明のデータ処理装置による BVMAP 命令の実行手順を示すフローチャートである。

*

【図 1】



* 【図 9 5】本発明のデータ処理装置による BVMAP 命令の実行に際してメモリのウェイト数が 0-0 である場合の外部バスサイクルを示すタイミングチャートである。

【図 9 6】本発明のデータ処理装置による BVMAP 命令の実行に際してメモリのウェイト数が 0-1 である場合の外部バスサイクルを示すタイミングチャートである。

【図 9 7】本発明のデータ処理装置による BVMAP 命令の実行に際してメモリのウェイト数が 0-2 である場合の外部バスサイクルを示すタイミングチャートである。

10 【図 9 8】本発明のデータ処理装置による BVMAP 命令の実行に際してメモリのウェイト数が 0-3 である場合の外部バスサイクルを示すタイミングチャートである。

【図 9 9】本発明のデータ処理装置による BVMAP 命令の実行時のデータ処理効率の各ウェイト数に対する一覧表である。

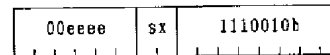
【図 1 0 0】本発明のデータ処理装置のストリング操作命令で指定可能な条件(eeee)とそのビットパターンとを示す一覧表である。

【符号の説明】

23	第 1 ALU(MALU)
27	第 2 ALU(SALU)
100	データ処理装置
101	アドレスバス
102	データバス
103	主記憶装置
150	命令フェッチ部
151	命令デコード部
153	アドレス生成部
154	PC生成部
30 156	オペランドアクセス部
157	バスインターフェイス部

【図 8】

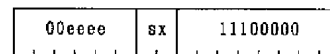
SMOV/eeee/b.sx



sx : エレメントのサイズ、割出し条件(R3)のサイズ
b : 処理方向
b=0 アドレス増加の方向に処理する(/P)
b=1 アドレス減少の方向に処理する(/B)
eeee : 割出し条件

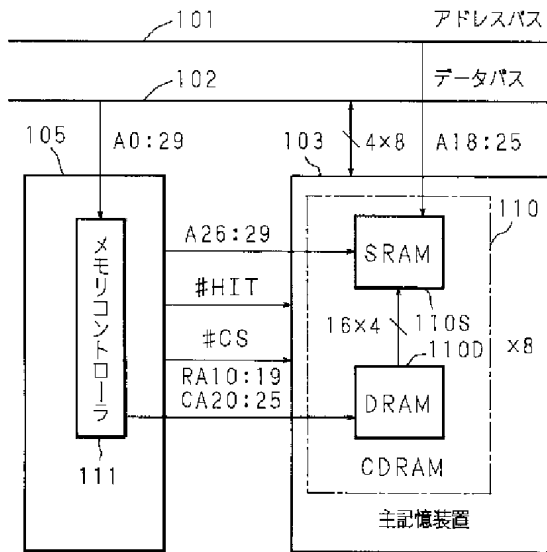
【図 9】

SCMP/eeee.sx



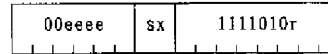
sx : エレメントのサイズ、割出し条件(R3)のサイズ
eeee : 割出し条件

【図2】



【図10】

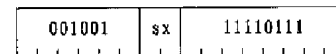
SSCH/eeee/r.sx



sx : エレメントのサイズ、割出し条件(R3)のサイズ
 eeee : 割出し条件
 r : ポインタ更新の方法
 r=0 エレメントサイズだけ増加(/F)
 r=1 増減値をレジスタR5で指定(/R)

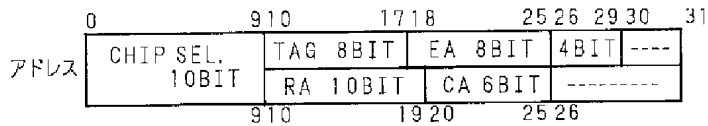
【図11】

SSTR.sx



sx : エレメントのサイズ、書き込みデータ(R3)のサイズ

【図3】

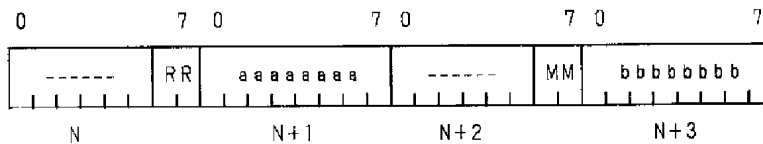


【図4】

〈通しのビット番号〉

0 7 8 1516 2324 31

〈バイト毎のビット番号〉



〈アドレス〉

←低アドレス→

高アドレス→

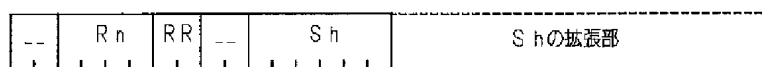
←MSB側

LSB側→

→→命令を読む方向→→

【図5】

BYTE: 0 1 2 ----- N+2-1



【図6】

BYTE: 0 1 2 N+2-1



【図7】

BYTE: 0 1 2 N+2-1

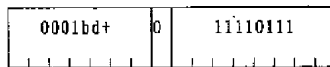


BYTE: N+2 N+2+1 N+2+2 N+2+M+2-1



【図12】

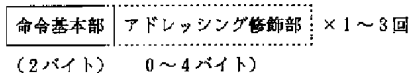
BVSCH/d/b



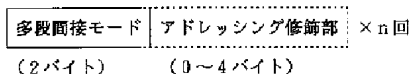
d : サーチするビット値
 d=0 "0"をサーチ(/0)
 d=1 "1"をサーチ(/1)
 b : サーチ方向
 b=0 ビット番号増加方向(/F)
 b=1 ビット番号減少方向(/B)
 R0 : base
 R1.s : offset
 read-modify-writeのオペランドになっており、パラメータである検索開始offsetとリターンパラメータである検索結果のoffsetが入る。dで指定した値のビットが見付かるまで、offsetは何回もワード境界を越える。命令中断時には、その時点で検索中のoffsetが入る。
 offset(R1)は符号付きとして扱われ、負の値も許される。
 R2.s : width
 offsetを検索するビットフィールドの長さ(ビット数)。
 width(R2)も符号付きとして扱われるが、width ≤ 0の場合は、V_flagのセットだけを行なってそのまま命令を終了する。

【図19】

・基本フォーマット

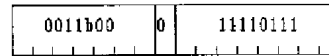


・アドレッシングモード拡張部のフォーマット



【図13】

BYMAP/b



b : 処理方向
 b=0 ビット番号増加方向(/F)
 b=1 ビット番号減少方向(/B)
 R0 : srcのビットフィールドのbase(src_base)
 R1.s : srcのビットフィールドのoffset(src_offset)
 符号付きとして扱われ、負の値も許される。
 R2.s : width
 演算するbitfieldの長さ(ビット数)width(R2)
 も符号付きとして扱われるが、width ≤ 0の場合は、何もせずに命令を終了する。
 R3 : destのビットフィールドのbase(dest_base)
 R4.s : destのビットフィールドのoffset(dest_offset)
 符号付きとして扱われ、負の値も許される。
 R5 : 演算の種類
 下位4bitを使用する。

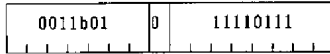
0000	F	False	0==>dest
0001	NAN	NotAndNot	#dest.and, #src==>dest
0010	AN	AndNot	dest.and, #src==>dest
0011	NS	NotSrc	#src==>dest
0100	NA	NotAnd	#dest.and, src==>dest
0101	ND	NotDest	#dest==>dest
0110	X	Xor	dest.xor, src==>dest
0111	NON	NotOrNot	#dest.or, #src==>dest
1000	A	And	dest.and, src==>dest
1001	NX	NotXor	#dest.xor, src==>dest
1010	D	Dest	dest==>dest
1011	ON	OrNot	dest.or, #src==>dest
1100	S	Src	src==>dest
1101	NO	NotOr	#dest.or, src==>dest
1110	O	Or	dest.or, src==>dest
1111	T	True	1==>dest

※ , and, は論理積を、, or, は論理和を # はビット反転を表す。

※ src=0, dest=0の演算結果を2^0のビットに、src=0, dest=1の演算結果を2^1のビットに、src=1, dest=0の演算結果を2^2のビットに、src=1, dest=1の演算結果を2^3のビットに、それぞれ入れている。

【図14】

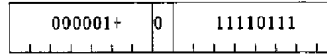
BVCPY/b



- b : 処理方向
 b=0 ビット番号増加方向(/F)
 b=1 ビット番号減少方向(/B)
- R0 : srcのビットフィールドのbase(src_base)
 R1.s : srcのビットフィールドのoffset(src_offset)
 符号付きとして扱われ、負の値も許される。
- R2.s : width
 演算するbitfieldの長さ(ビット数)
 width(R2) も符号付きとして扱われるが、
 width ≤ 0 の場合は、何もせずに命令を終了する。
- R3 : destのビットフィールドのbase(dest_base)
 R4.s : destのビットフィールドのoffset(dest_offset)
 符号付きとして扱われ、負の値も許される。

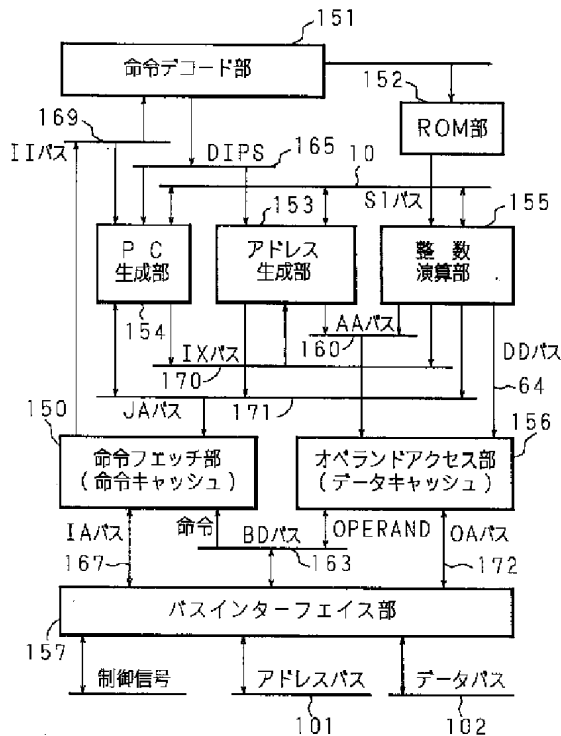
【図15】

BVPAT.s

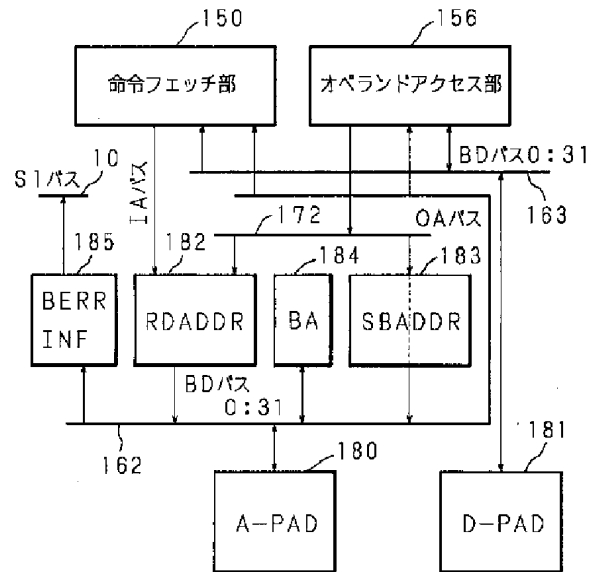


- R0.s : pattern
 R1 : 使用しない
 R2.s : width
 演算するbitfieldの長さ(ビット数)
 width(R2) は符号付きとして扱われるが、
 width ≤ 0 の場合は、何もせずに命令を終了する。
- R3 : destのビットフィールドのbase(dest_base)
 R4.s : destのビットフィールドのoffset(dest_offset)
 符号付きとして扱われ、負の値も許される。
- R5 : 演算の種類
 下位4bitを使用。BVMAP命令と共通。

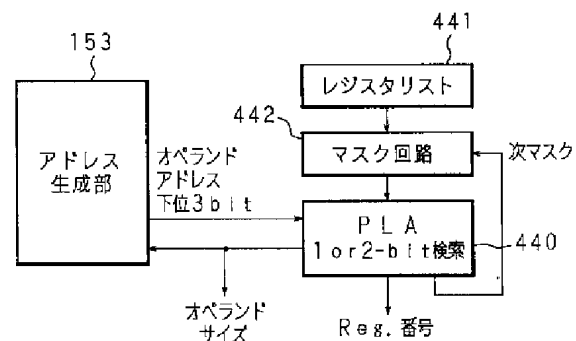
【図16】



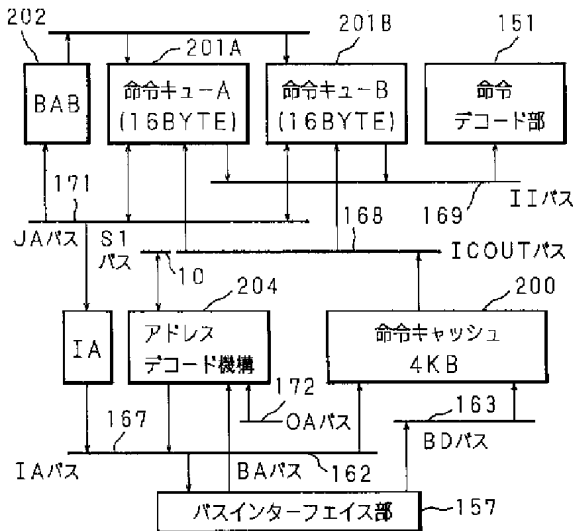
【図17】



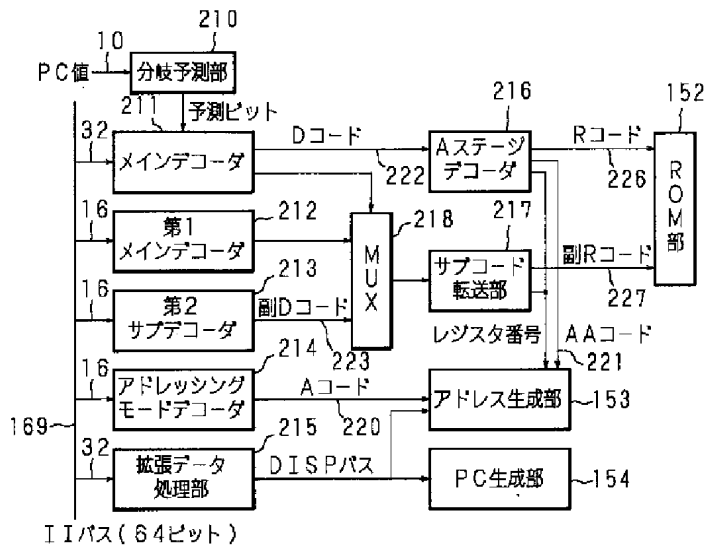
【図34】



【図18】



【図20】



【図39】

命令	性能(ノックアウト)	備考
BVSCI	32 bit	
BVMP	32 bit 64/3 bit 16 bit	R5=R, T R5=NS, S, ND, D R5=NA, AN, NA, X, NON, A, NX, ON, NO, O
BVCPY	64/3 bit	
BVPAT	32 bit 64/3 bit	R5=R, T, NS, S R5=NA, AN, NA, ND, X, NON, A, NX, D, ON, NO, O

【図35】

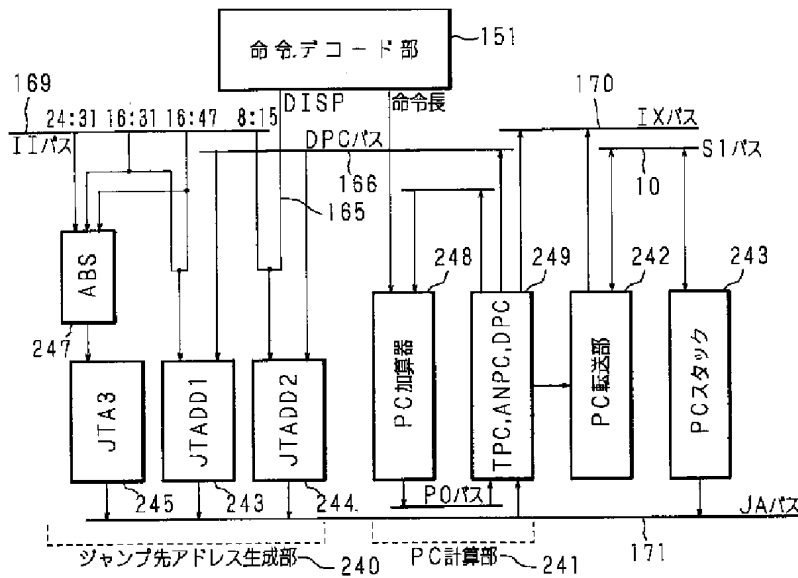
レジスタリスト	アドレス下位	転送方法
連続	B'100	第2オペランドから 2オペランドずつ転送
連続	B'100以外(*)	第1オペランドから 2オペランドずつ転送
非連続	don't care	つねに1オペランド ずつ逐次転送

*: B'000が望ましい。B'000以外はメモリ
アクセスが非整値で転送効率が悪い

【図38】

命令	性能(ノックアウト)	備考
SNDV	8/8 byte 2 byte 1/8 element	eeee=Nまたは、eeee=BQかつR3=0 eeee=NEまたは、eeee=BQかつR3≠0 eeee=N, BQ, NE以外
SCMP	8/8 byte 2 byte 1/8 element	eeee=Nまたは、eeee=BQかつR3=0 eeee=NEまたは、eeee=BQかつR3≠0 eeee=N, BQ, NE以外
SSCH	8/8 byte 1/8 element	eeee=N, BQ, NE eeee=N, BQ, NE以外
SSTR	4 byte	

【図21】



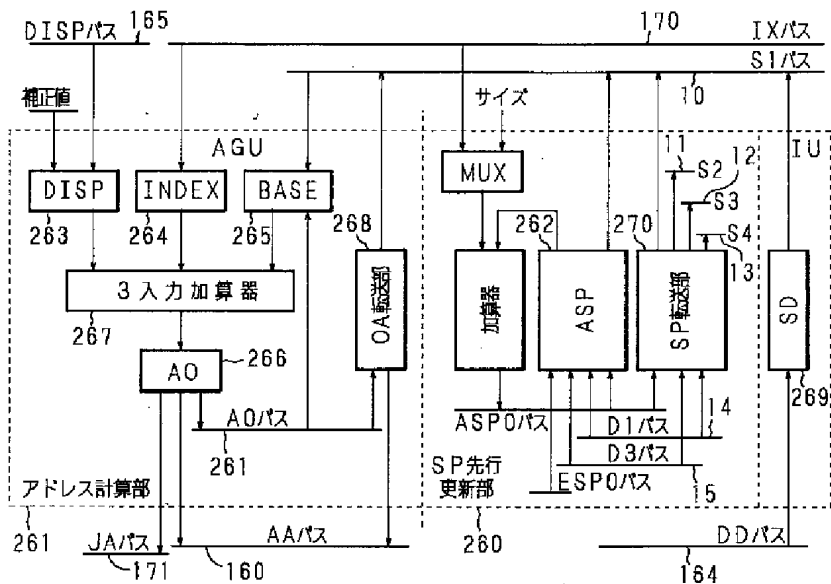
【図64】

MOV #0, R0
BSR Func1
MOV @R0, R1 ; 0番地へのメモリアクセス
は行なわれない。
Func1: MOV @R5, disp, R0
RTS

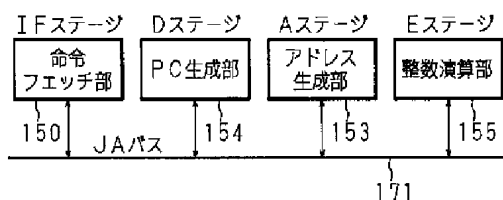
【図66】

CMP R0, R5
BGT L1
JSR @R6 ; BGTの条件確定までジャンプ
先アドレス計算を抑止。
L1: JSR @R7 ; BGTの条件確定までジャンプ
先アドレス計算を抑止。

【図22】



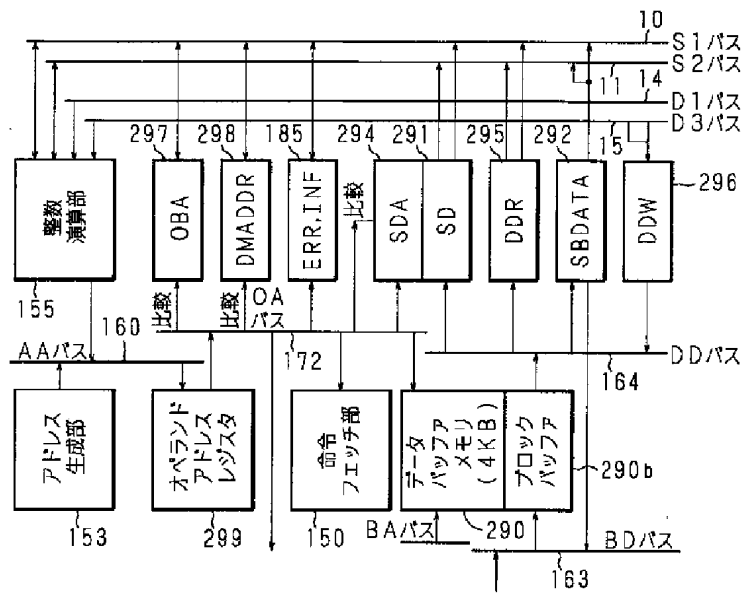
【図41】



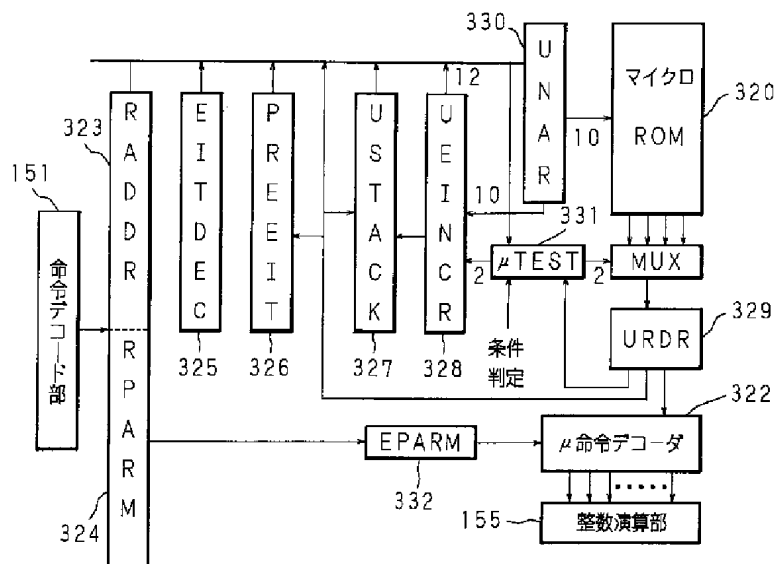
【図65】

CMP R0, R5
BGT L1
MOV @R0, R4 ; BGTの条件確定まで外部メモリ
のR0番地アクセスは抑止。
ADD R2, R0
MOV @R0, R1 ; BGTの条件確定まで外部メモリ
のR0番地アクセスは抑止。

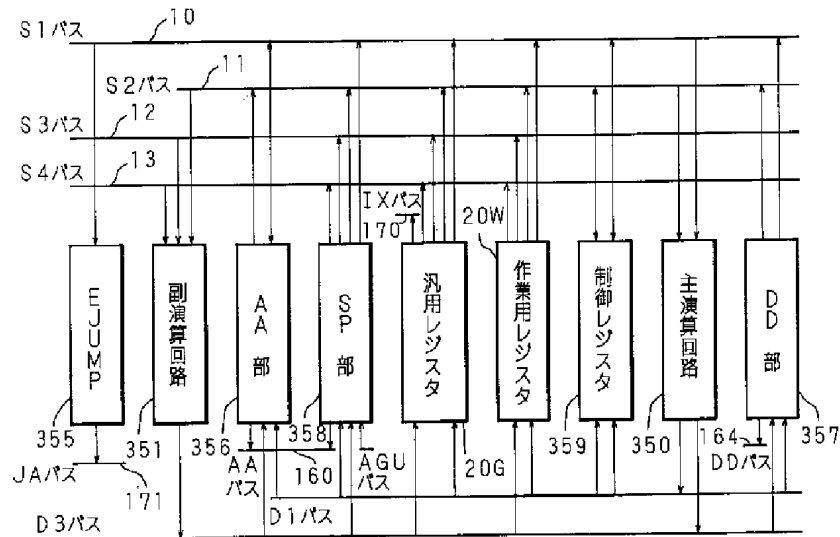
【図23】



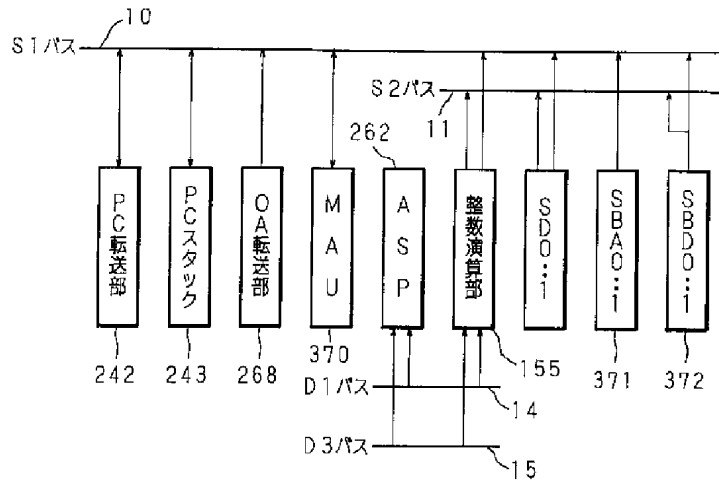
【図24】



【図25】

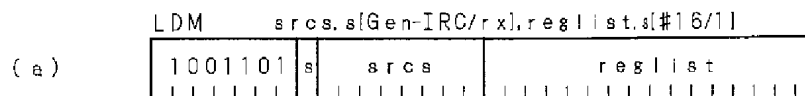


【図26】

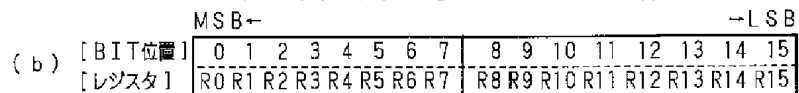


【図32】

命令のビットパターン



レジスタリスト (reglist) とレジスタ番号の対応



1 CLOCK CYCLE

TIME

处理命令

MOV R3,R1

SHL #2,R3

ADD R3,R2

ADD R5,R0

MOV @SP+,R4

ADD R3,R0

ADD #1,@(-4,R2)

MOV R1,R2

IF D A F E

IF D A F E

IF - D A F E

IF - D A F E

IF - D A F E

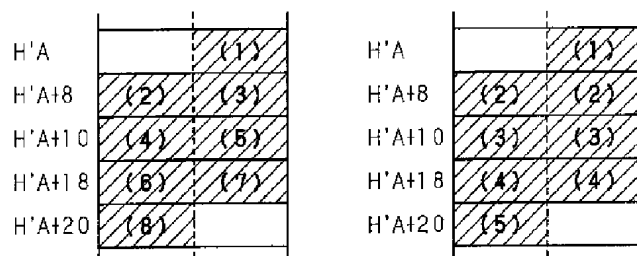
IF - D A F E

IF - - D A F E

- : PIPELINE STALL

/// : DATA CACHE ACCESS

(b) G400の
2オペランド同時転送



【図29】

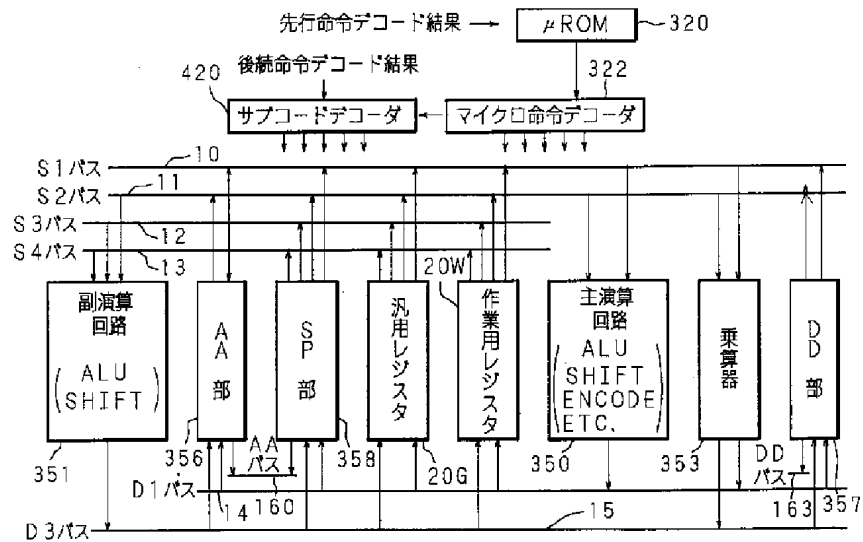
先行命令の種類（命令全体または最終処理部）	0	15	16	31	32	47
(1) 16BITの命令基本部のみ	OP_CODE		OP_CODE			
(2) 16BITの命令基本部+16BITの拡張部	OP_CODE		exp		OP_CODE	
(3) 16BITの命令基本部が2つ連続	OP_CODE		OP_CODE		OP_CODE	
(4) 16BITの拡張部	OP_CODE		OP_CODE			
(5) 32BITの拡張部	exp				OP_CODE	

OP_CODE : 先行命令の基本部

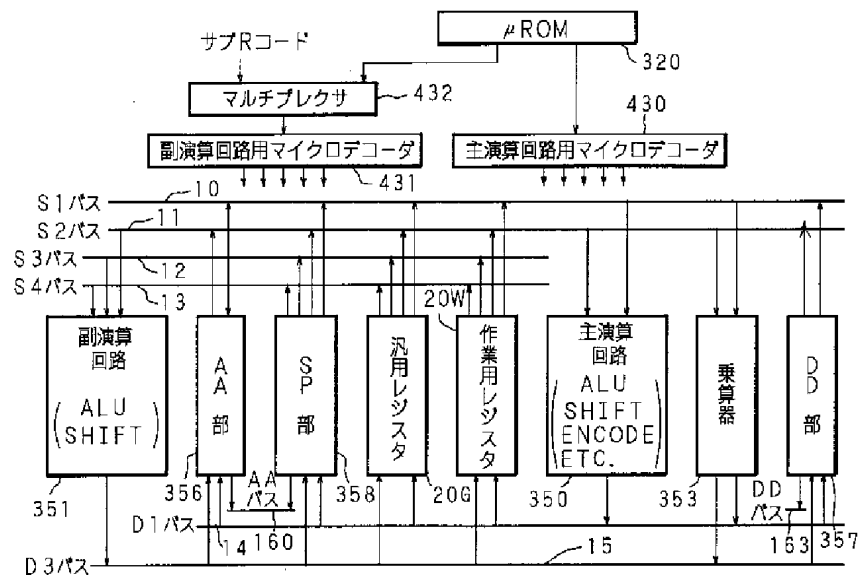
exp : 先行命令のアドレッシング拡張部

OP_CODE : 後続命令の基本部

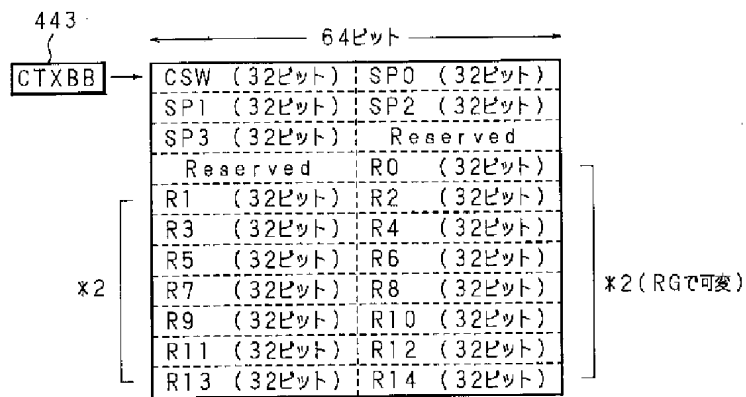
【図30】



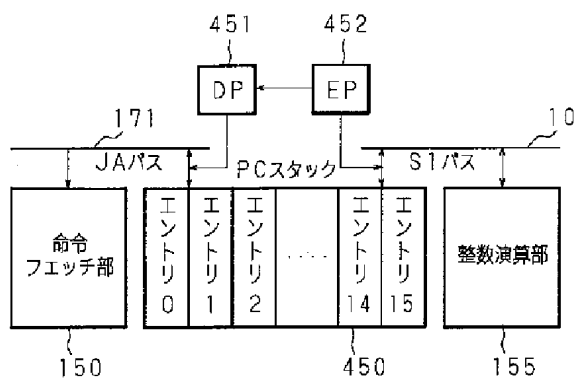
【図31】



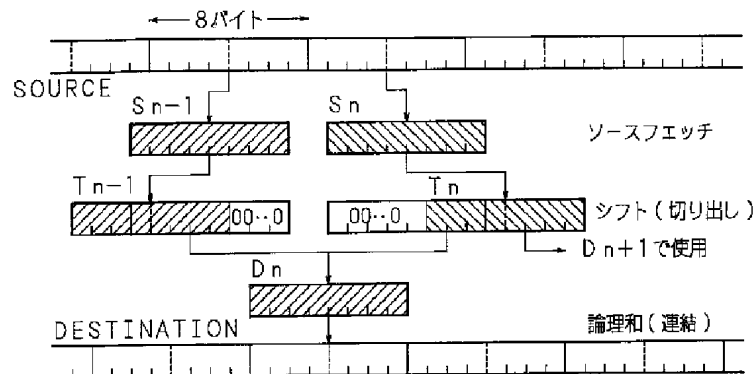
【図36】



【図47】



【図37】

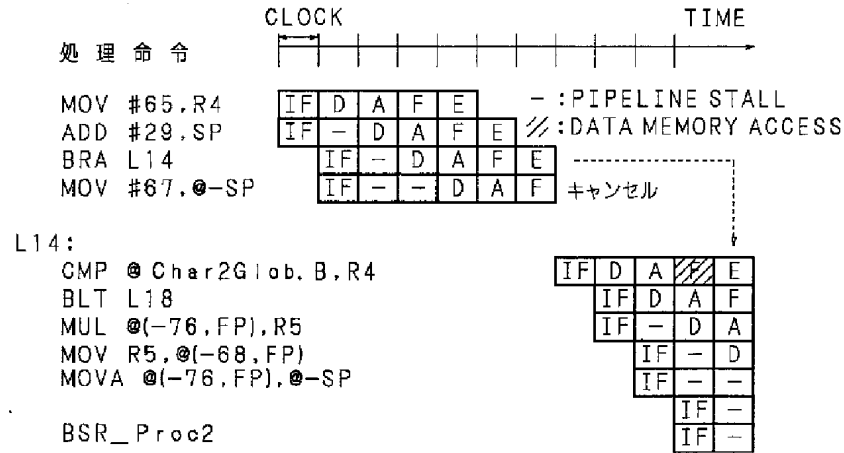


【図40】

ステージ	処理するジャンプ命令	ジャンプ命令のためのハードウェア
IFステージ	なし	2系列の命令キュー
Dステージ	BRA, Bcc, BSR, ACB, SCB, RTS, EXITD, JP, JSRの一部ロッキング	2つの分岐先アドレス加算器 分岐予測テーブル、PCスタック
Aステージ	Dステージジャンプ以外の JP, JSR	アドレス加算器 (オペランドアドレス加算器を兼用)
Fステージ	なし	なし
Eステージ	分岐予測の外はBcc, ACB, SCB, RTS, EXITD	ALLなどの整数演算機構
Sステージ	なし	なし

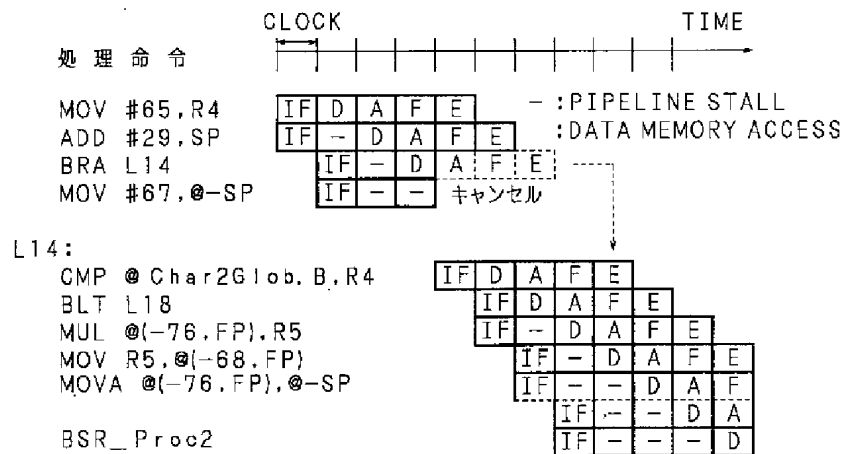
【図42】

命令実行ステージでBRA命令のジャンプ処理を行う場合

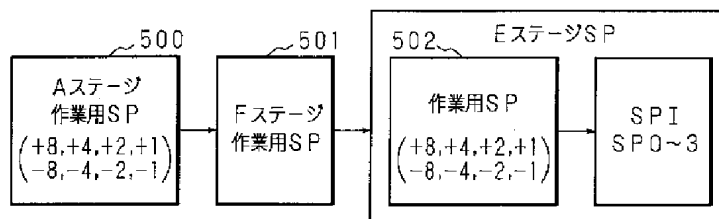


【図43】

命令デコードステージでBRA命令の先行ジャンプ処理を行う場合



【図57】



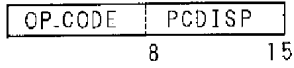
【図44】

命令	先行ジャンプ	処理時間	コメント
BRA	Dステージ	2	
Bcc ;Case1	Dステージ	2	履歴による予測分岐を示すとき 予測誤った時は処理時間4クロック
Bcc ;Case2	行わない	1	履歴による予測分岐を示すとき 予測誤った時は処理時間4クロック
ACB QR d, B	Dステージ	2	ジャンプしない時は処理時間7クロック
ACB GE d, B	Dステージ	4	ジャンプしない時は処理時間8クロック
ACB その他	Dステージ	3	ジャンプしない時は処理時間8クロック
SCB QR d, B	Dステージ	2	ジャンプしない時は処理時間7クロック
SCB GE d, B	Dステージ	4	ジャンプしない時は処理時間8クロック
SCB その他	Dステージ	3	ジャンプしない時は処理時間8クロック
JMP eaddr1	Dステージ	2	@abs, @PC, disp:32)のみ
JMP eaddr2	Aステージ	3	多段モード指定は更に処理時間必要
JSR eaddr1	Dステージ	2	@abs, @PC, disp:32)のみ
JSR eaddr2	Aステージ	3	多段モード指定は更に処理時間必要
BSR	Dステージ	2	
RTS	Dステージ	2	リターンが誤った場合6クロック必要
ENTER:G	ジャンプなし	6 + 2m	m: 退避するレジスタ数
ENTER:E	ジャンプなし	5 + 2m	m: 退避するレジスタ数
EXITD:G	Dステージ	9 + m	m: 復帰するレジスタ数, PCスタックヒット時
EXITD:E	Dステージ	7 + m	m: 復帰するレジスタ数, PCスタックヒット時
NOP	ジャンプなし	1	
FBcc	行わない	8	浮動小数点フラグによる条件分岐
FNOP	行わない	1	
COP_JCC	行わない	7	コプロセッサ条件ジャンプ

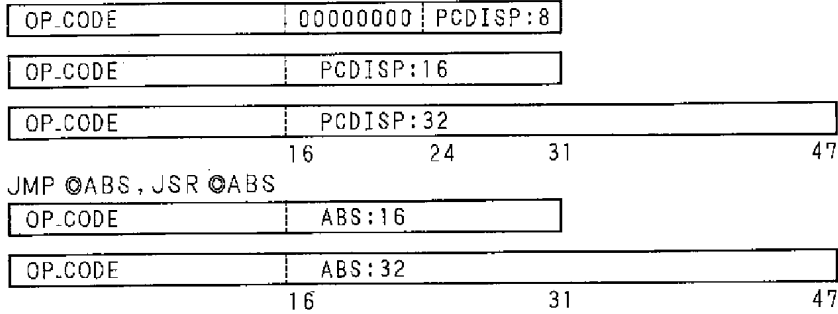
(注) QR d, B : ACB, SCB 命令で R フォーマットかつ pcdisp が 8 ビットのもの
 GE d, B : ACB, SCB 命令で GE フォーマットかつ pcdisp が 8 ビットのもの
 その他 : ACB, SCB 命令で (QR d, B), (GE d, B) 以外のもの
 eaddr1 : 絶対アドレスまたはオフセットが 32 ビットの PC 相対アドレス
 eaddr2 : 絶対、オフセットが 32 ビットの PC 相対以外のアドレス

【図45】

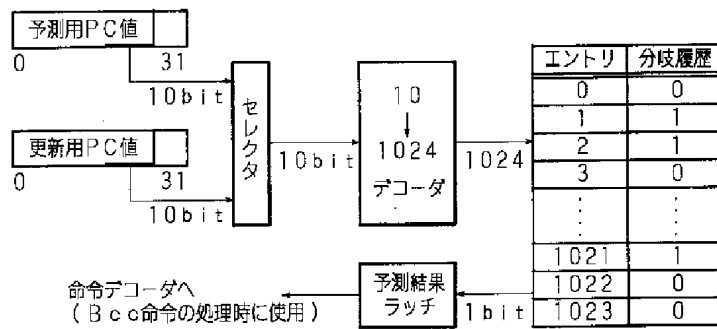
BRA:D, BCC:D, BSR:D



BRA:D, BCC:G, BSR:G, JMP ©(PC, PCDISP:32), JSR ©(PC, PCDISP:32), ACBとSCBの一部



【図46】

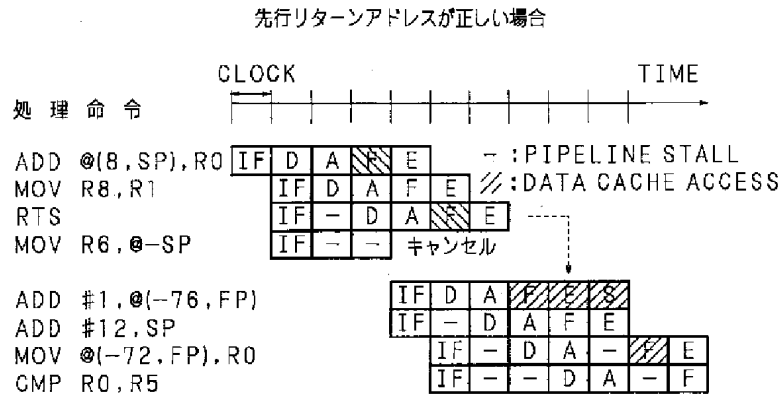


【図63】

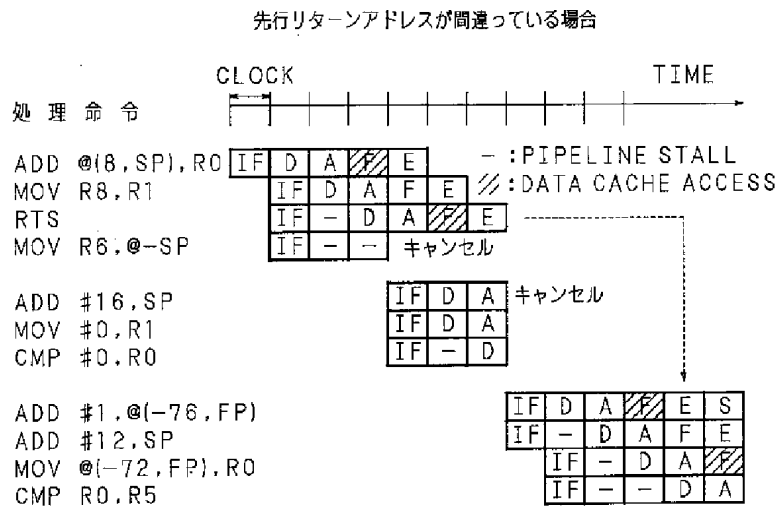
先行命令のオペランドストア先	後続命令のストア先	ストール時間
レジスタ	don't care	0 clock cycle
メモリ(8バイトクロスなし)	レジスタ	0 clock cycle
	メモリ	1* clock cycle
メモリ(8バイトクロスあり)	レジスタ	0 clock cycle
	メモリ	2* clock cycle

*:ストアバッファがfullのときはストール時間がさらに長。

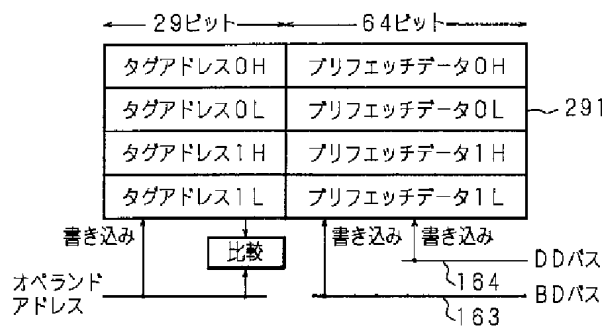
【図48】



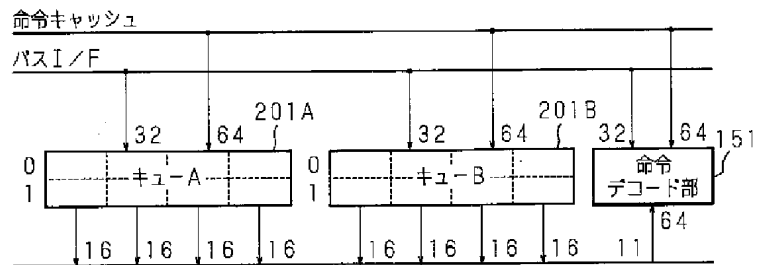
【図49】



【図59】

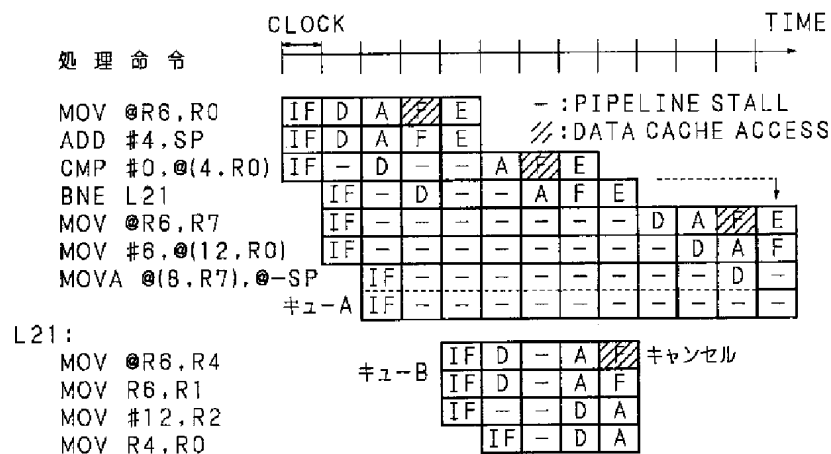


【図50】



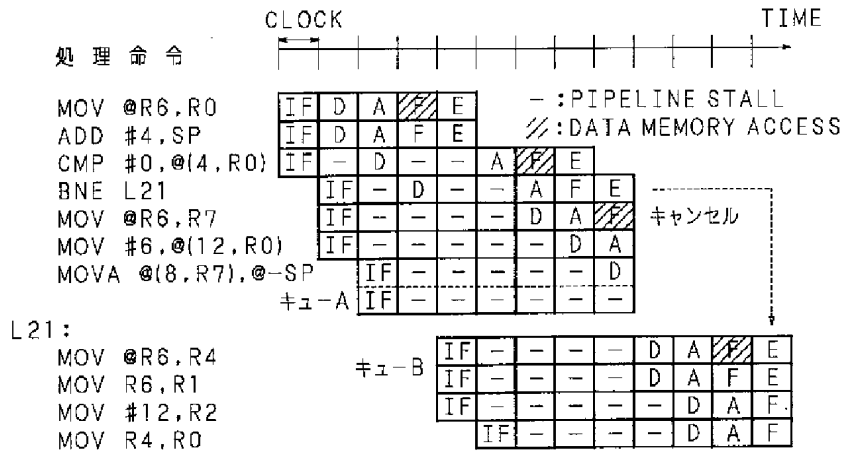
【図51】

非分岐の条件分岐命令に対して先行分岐した場合

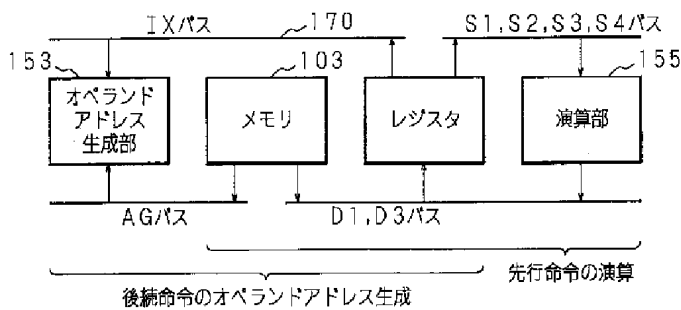


【図52】

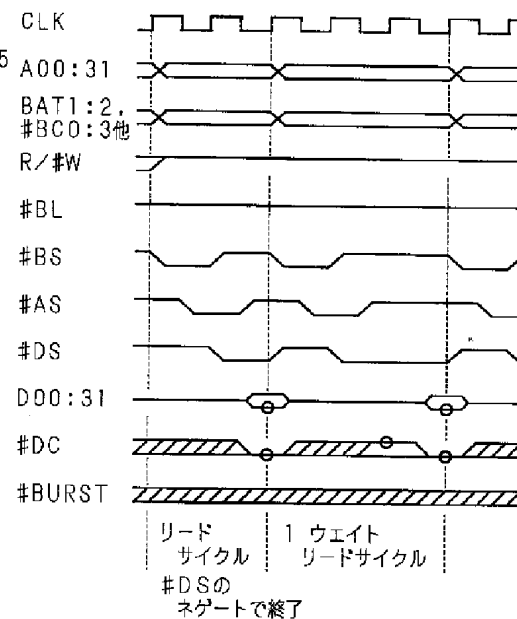
分岐する条件分岐命令に対して先行分岐をしなかった場合



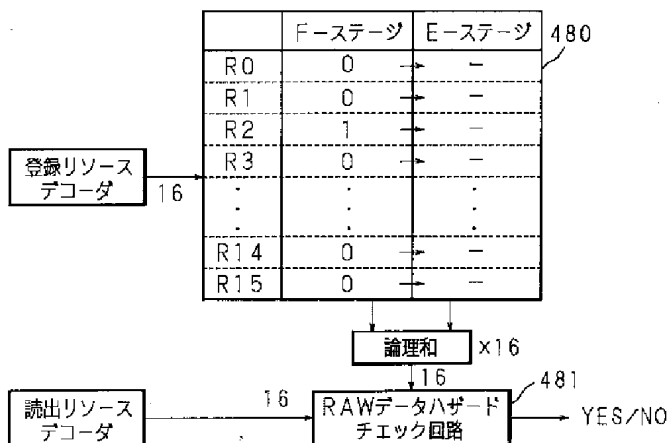
【図53】



【図68】

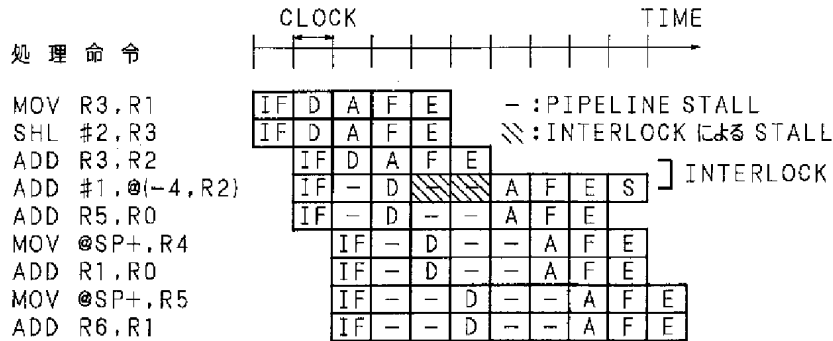


【図54】



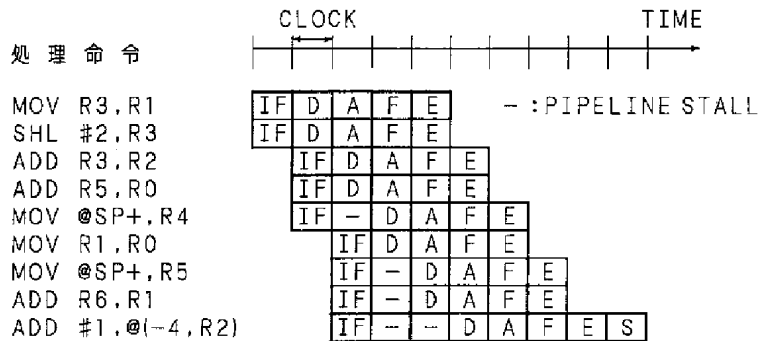
【図55】

ハイラインインターロックによるストールがある場合

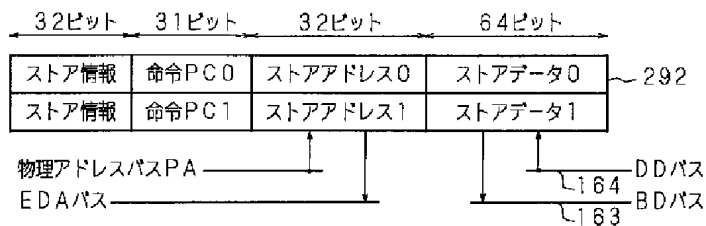


【図56】

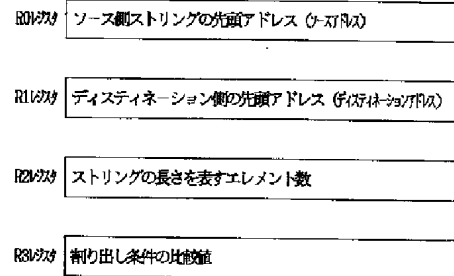
命令の順序を入れ換えてパイプラインインターロックを排除した場合



【図60】



【図80】

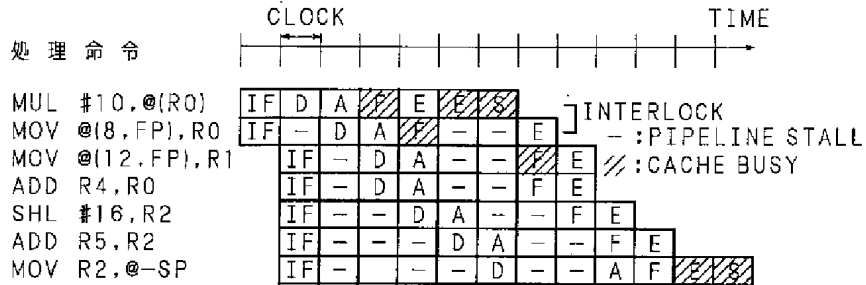


【図58】

先行命令destバリエーション	後続命令のアドレッシングモード	ストール
Rm m=0~14	@(Rn), @(exp, Rn) n=m	あり
	@(Rn), @(exp, Rn) n≠m	なし
	Rn, @abs, #imm, @SP+, @-SP	なし
	@(exp, PC), @(PC:b, exp, Rn)	なし
	@(Rb:B, exp, Rx) b=m	あり
	@(Rb:B, exp, Rx) x=m	あり
	@(Rb:B, exp, Rx) b≠m かつ X≠m	なし
	@@(Rb:B, exp, Rx) b=m	あり
	@@(Rb:B, exp, Rx) x=m	あり
	@@(Rb:B, exp, Rx) b≠m かつ X≠m	なし
R15	@(R15), @(exp, R15)	あり
	@(Rn), @(exp, Rn) n≠15	なし
	@SP+(POPのsrc, RTSを含む)	あり
	@-SP (PUSHのdest, BSR, JSRを含む)	あり
	Rn, @abs, #imm	なし
	@(exp, PC), @(PC:b, exp, Rn)	なし
	@(R15:B, exp, Rx)	あり
	@(Rb:B, exp, R15)	あり
	@(Rb:B, exp, Rx) b≠15かつ X≠15	なし
	@@(R15:B, exp, Rx)	あり
	@@(Rb:B, exp, R15)	あり
	@@(Rb:B, exp, Rx) b≠15かつ X≠15	なし

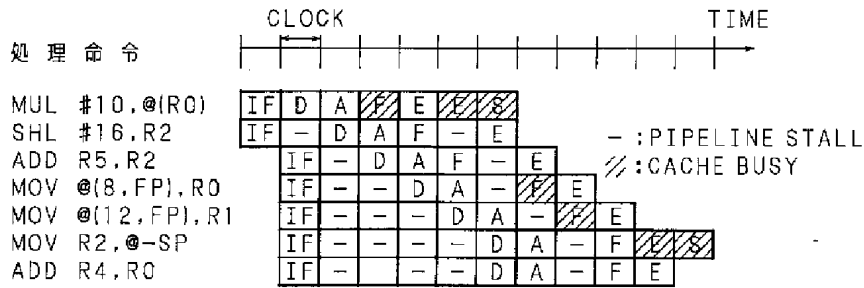
【図61】

ハイブラインインターロックによるストールがある場合

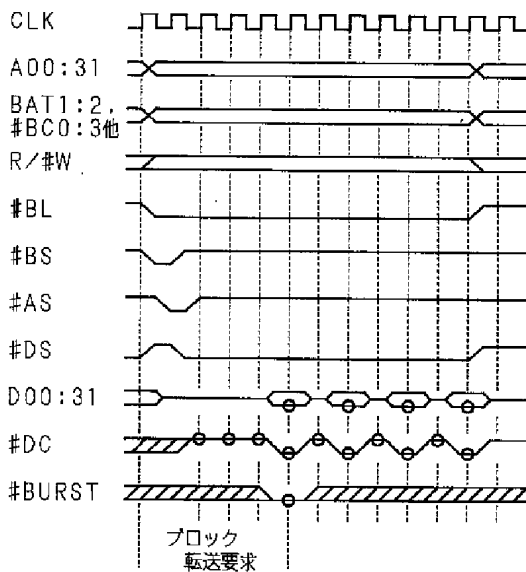


【図62】

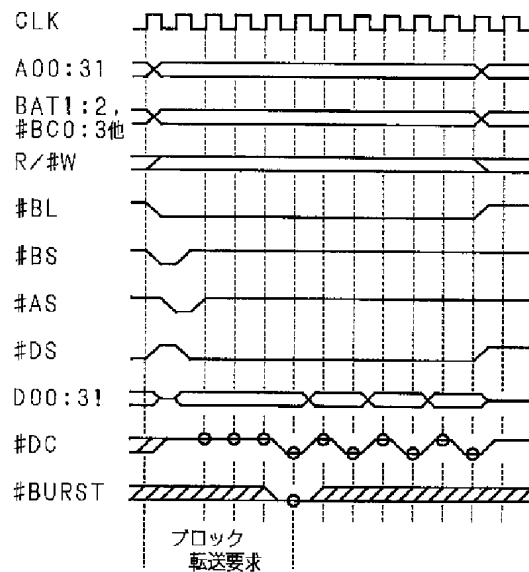
命令の順序を入れ換えてパイプラインインターロックを排除した場合



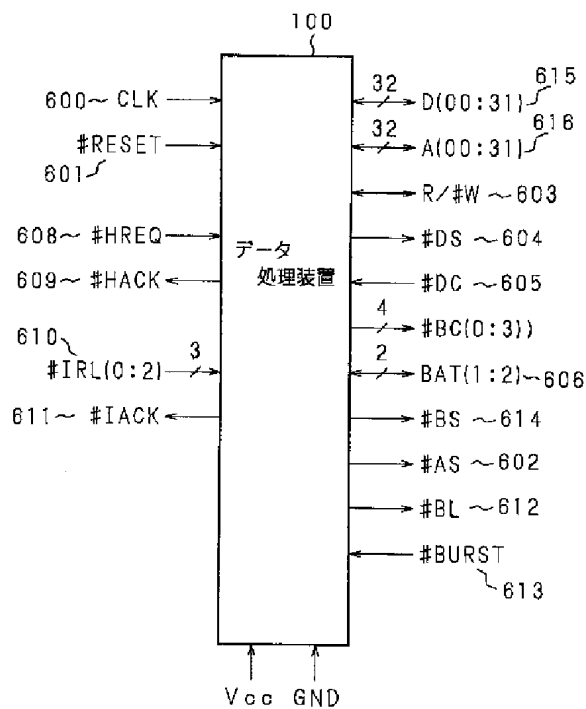
【図71】



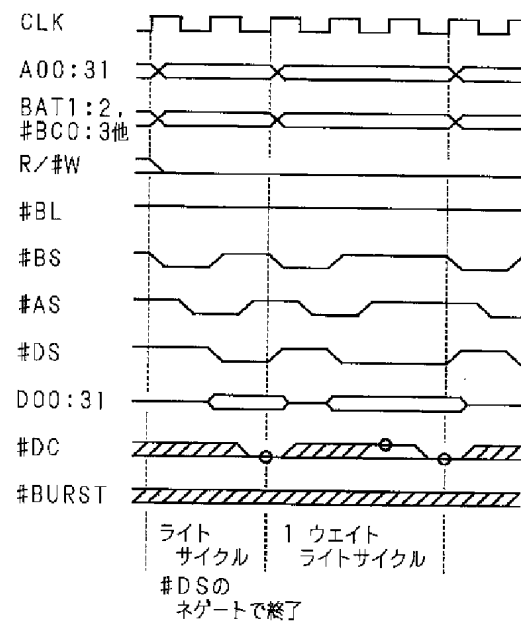
【図73】



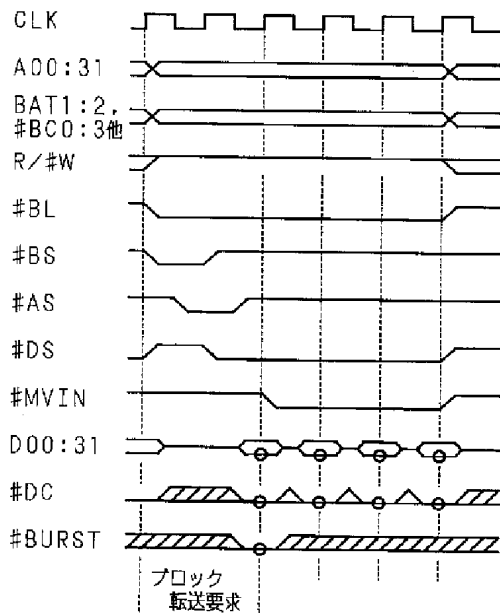
【図67】



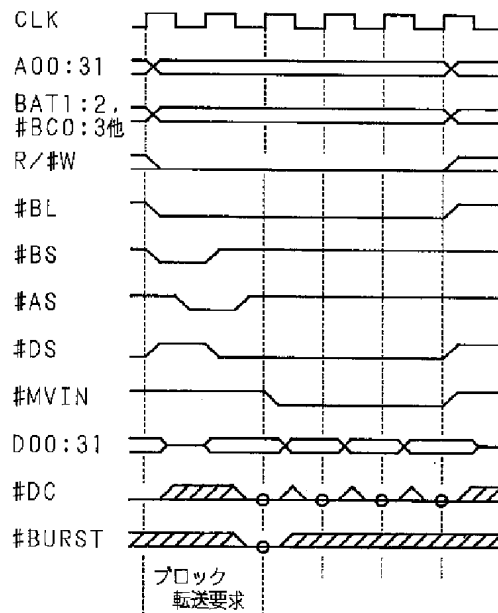
【図69】



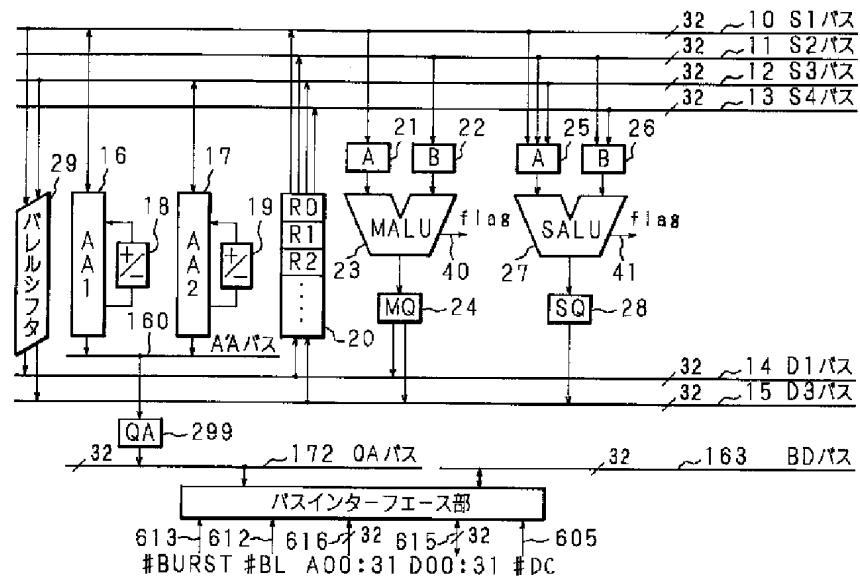
【図70】



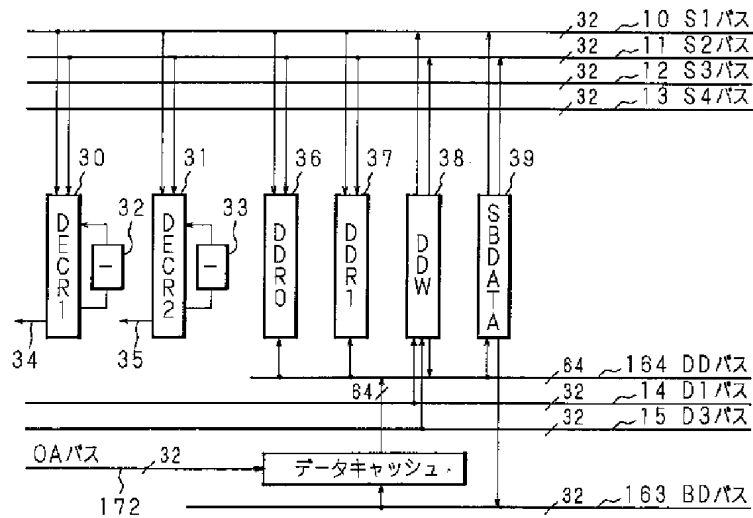
【図72】



【図74】



【図75】



【図88】

外部メモリのウェイト数	SMDY命令のデータ処理効率
0-0	18.8ビット/クロック
1-0	18.8ビット/クロック
2-0	16ビット/クロック
8-0	14.2ビット/クロック

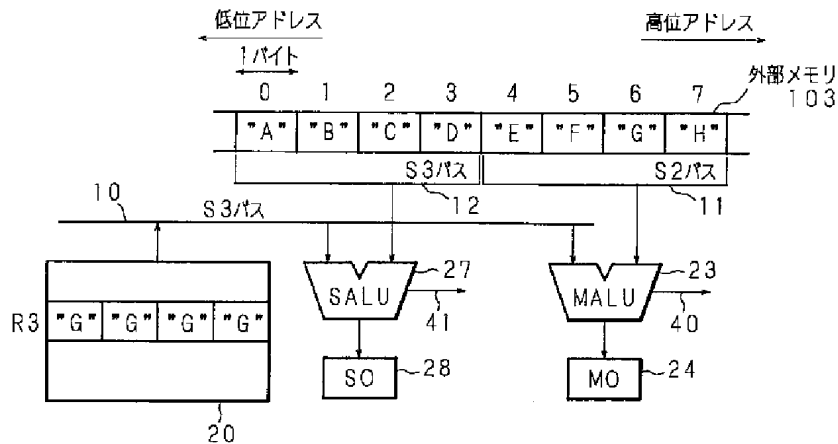
【図76】

フィード名	オペレーション	機 能
maai	s1 0	s1バスの値を入力する ゼロクリア
mabi	s2 0	s2バスの値を入力する ゼロクリア
maop	+ +1 - -1 \$ \$1 and #and and# or #or or# eopf_e eopf_n eopb_e eopb_n	A + B A + B + 1 A - B A - B - 1 B - A B - A - 1 A and B #A and B A and #B A or B #A or B A or #B 順方向の一致検出演算を行なう 順方向の不一致検出演算を行なう 逆方向の一致検出演算を行なう 逆方向の不一致検出演算を行なう
masu	s u	符号拡張する ゼロ拡張する
maaz	1 2 4 oz	1バイト 2バイト 4バイト ozレジスタの値をサイズとして使用する
mabz	1 2 4	1バイト 2バイト 4バイト
mafz	1 2 4	1バイト 2バイト 4バイト
maot	d1 d3 d1 d3 x	d1バスに出力する d3バスに出力する d1バスとd3バスに出力する 出力しない

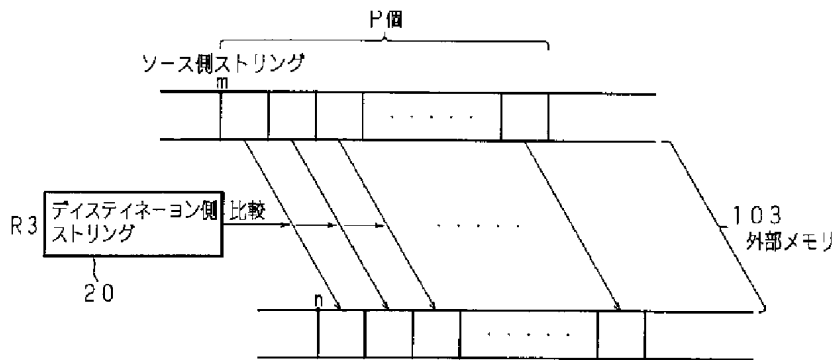
【図77】

フィード名	オペレーション	機 能
saa i	s1 s2 s3 0	s1バスから入力する s2バスから入力する s3バスから入力する ゼロクリアする
sabi	s2 s4 0	s2バスから入力する s4バスから入力する ゼロクリアする
saop	+ +1 - -1 \$ \$1 and #and and# or #or or# eopf_e eopf_n eopb_e eopb_n	A + B A + B + 1 A - B A - B - 1 B - A B - A - 1 A and B #A and B A and #B A or B #A or B A or #B 順方向の一致検出演算を行なう 順方向の不一致検出演算を行なう 逆方向の一致検出演算を行なう 逆方向の不一致検出演算を行なう
sasu	s u	符号拡張する ゼロ拡張する
saaz	1 2 4 oz	1バイト 2バイト 4バイト ozレジスタの値をサイズとして使用する
sabz	1 2 4	1バイト 2バイト 4バイト
safz	1 2 4	4バイト 2バイト 4バイト

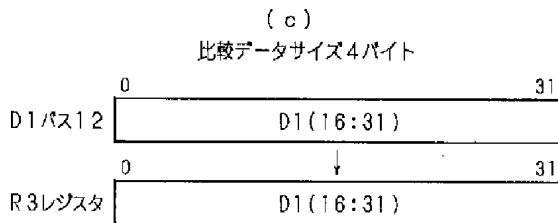
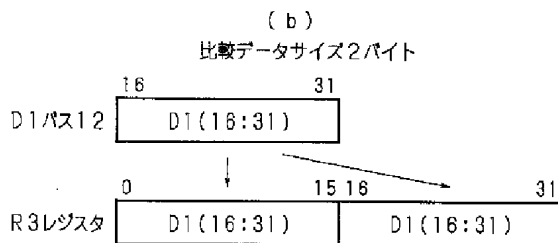
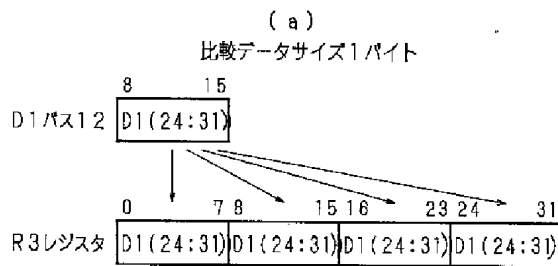
【図78】



【図79】



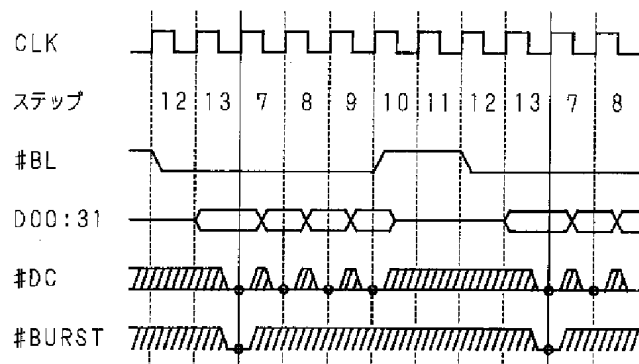
【図81】



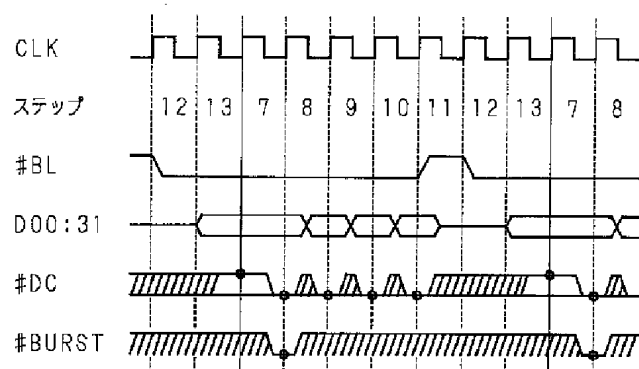
【図99】

外部メモリのウェイト数	BVMAP命令のデータ処理効率
0-0	16ビット/クロック
1-0	16ビット/クロック
2-0	16ビット/クロック
3-0	14.2ビット/クロック

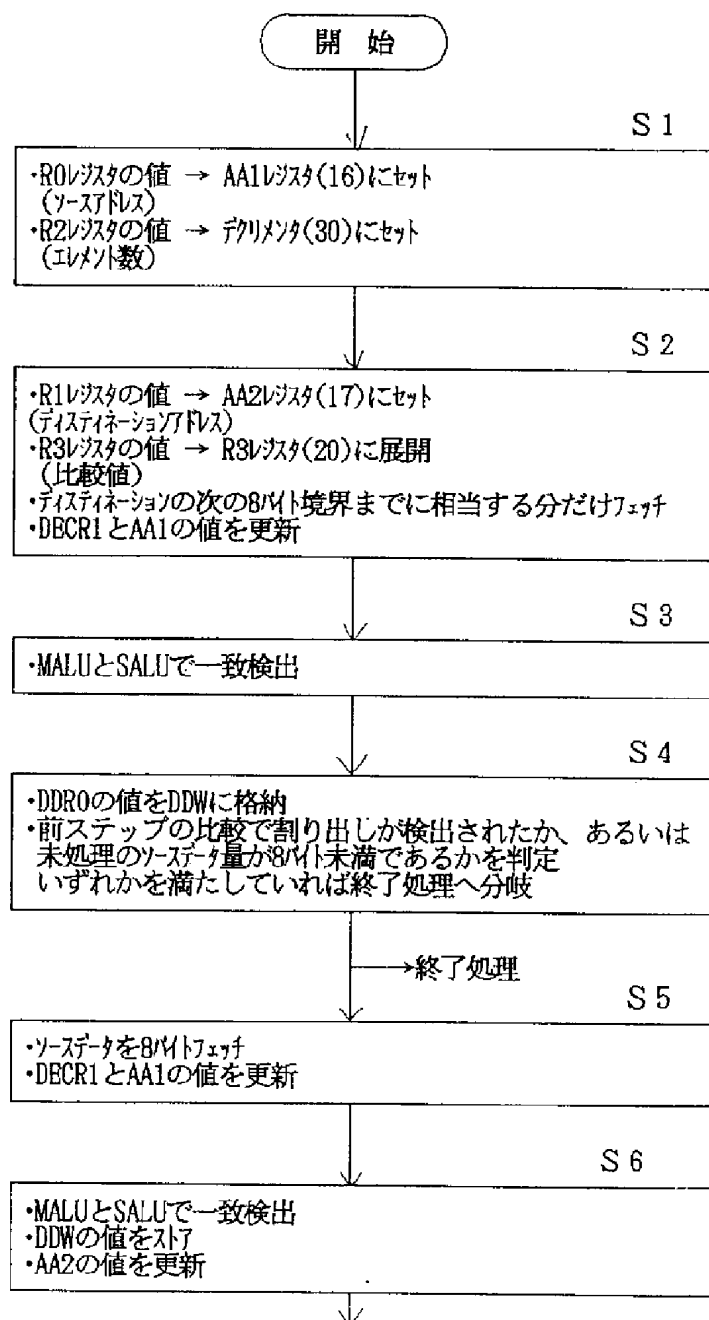
【図84】



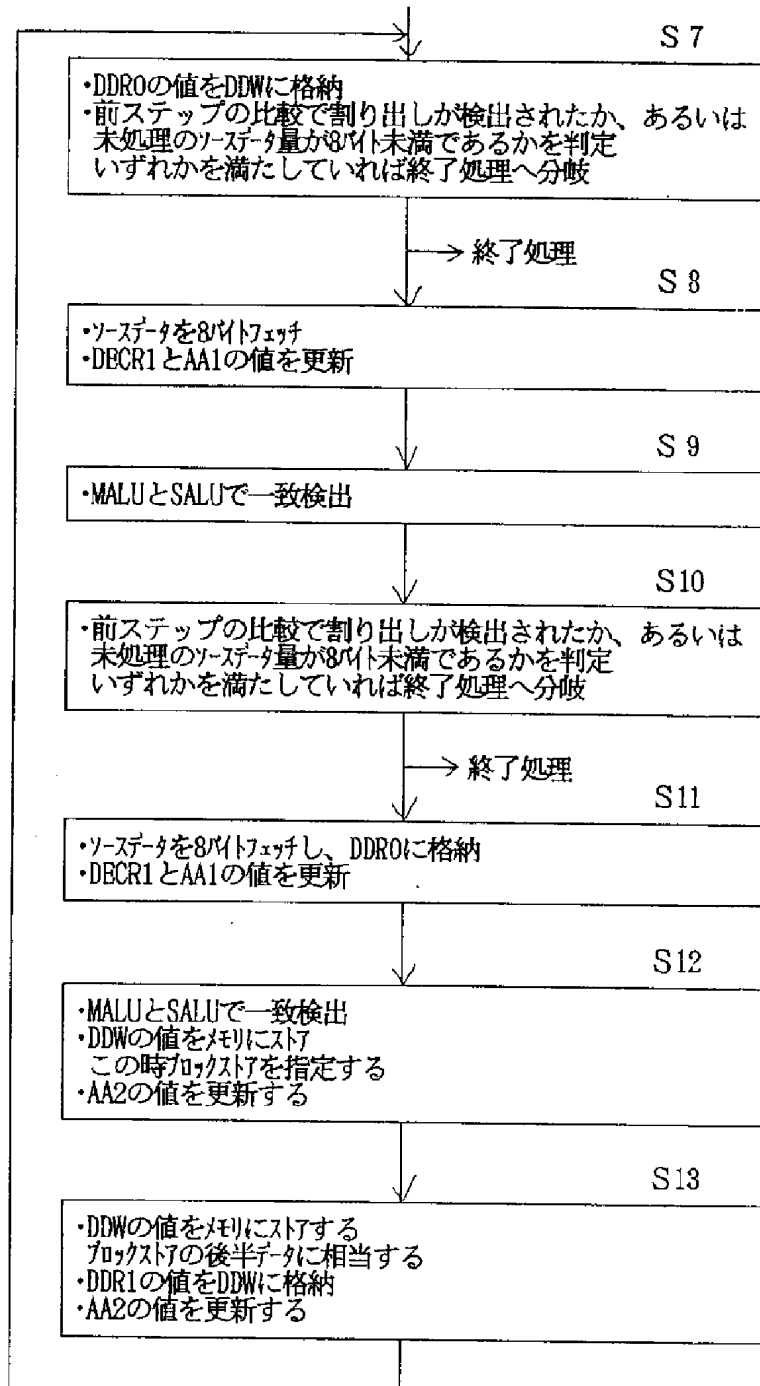
【図85】



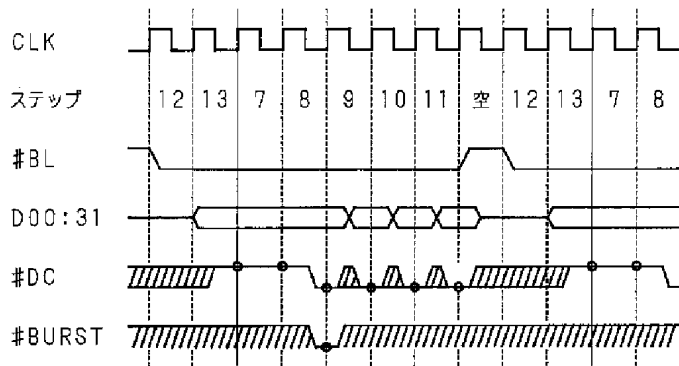
【図82】



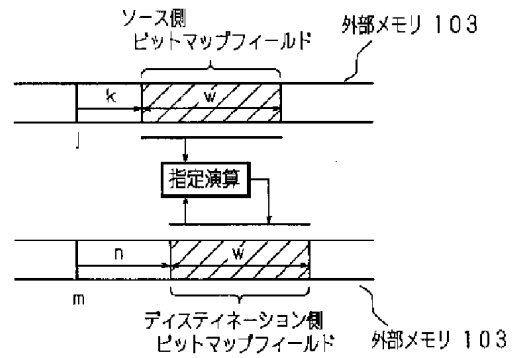
【図83】



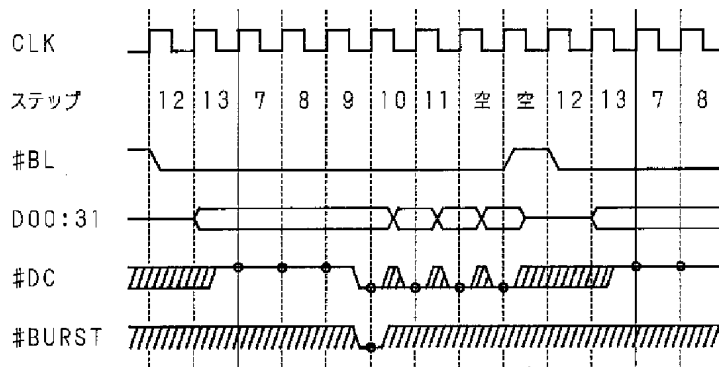
【図86】



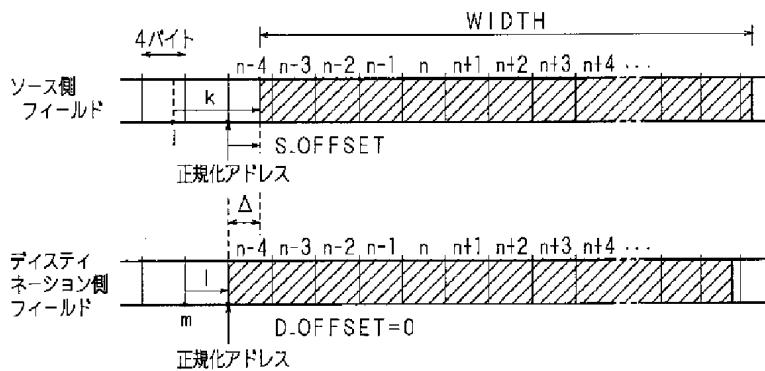
【図89】



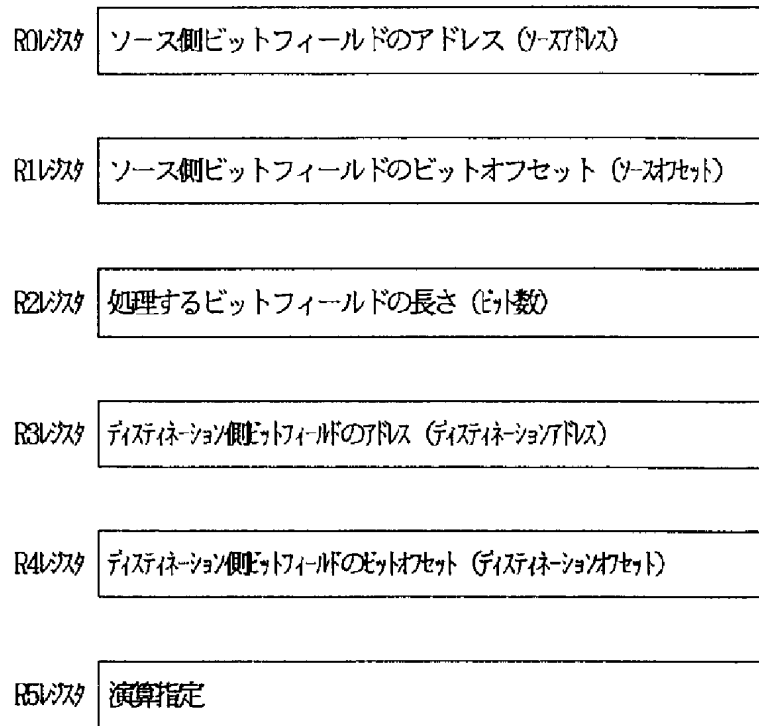
【図87】



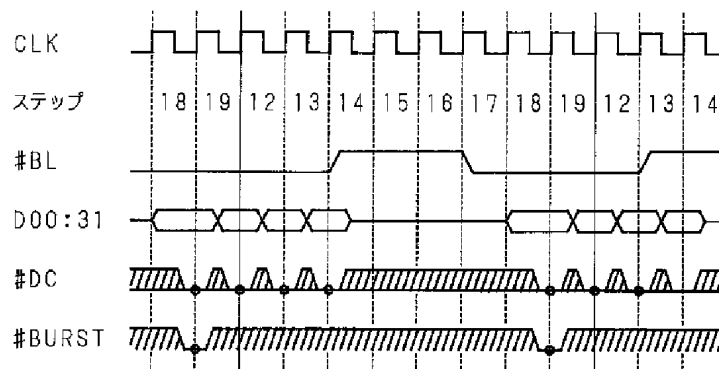
【図91】



【図90】



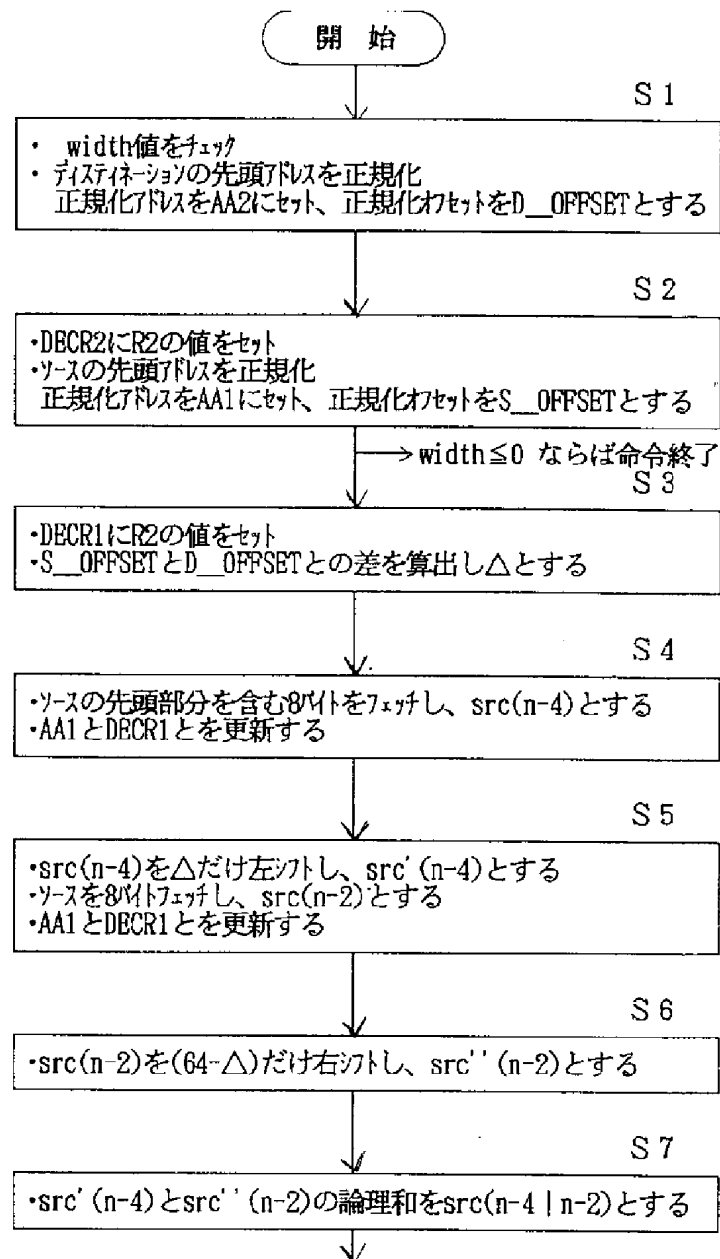
【図95】



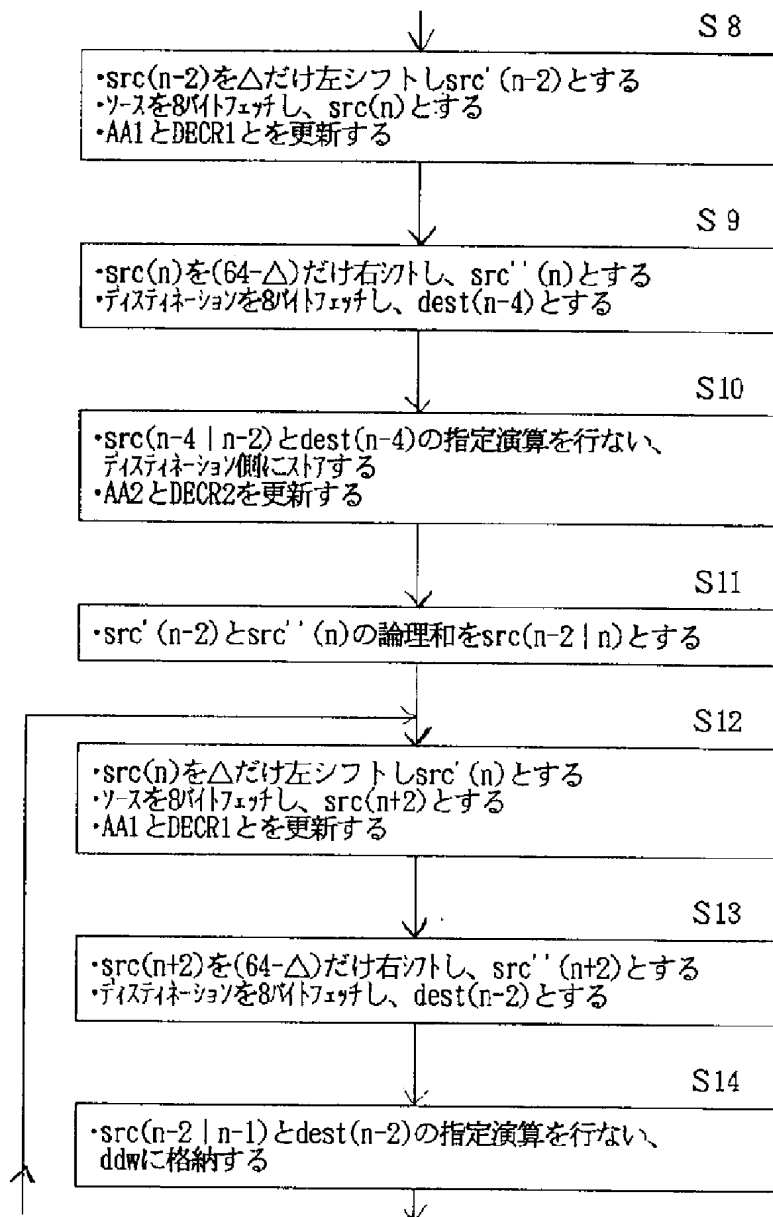
【図100】

前出し条件=終了条件 (=検査条件) (=処理続行条件の逆)	オプションのニーモニック	eeee
< R3	LTU :less than (unsigned)	0000
= R3	GEU :greater or equal (unsigned)	0001
> R3	BQ :equal	0010
< R3	NE :not equal	0011
< R3	LT :less than (signed)	0100
= R3	GE :greater or equal (signed)	0101
前出し条件なし	N :never(またはオプションなし)	0110

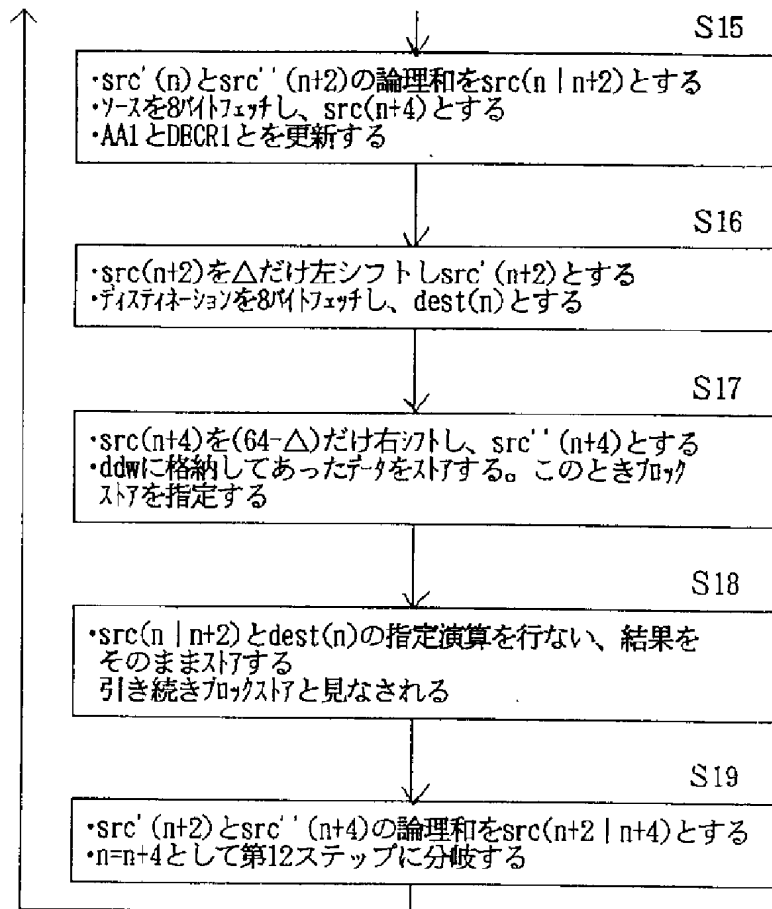
【図92】



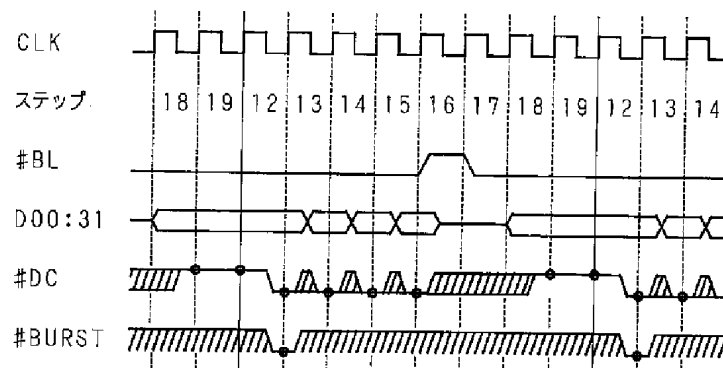
【図93】



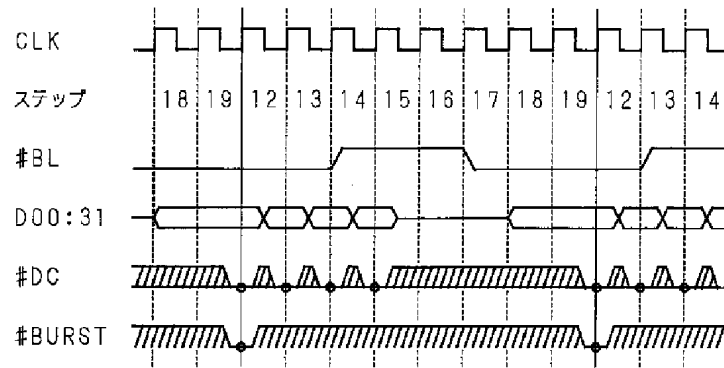
【図94】



【図97】



【図96】



【図98】

